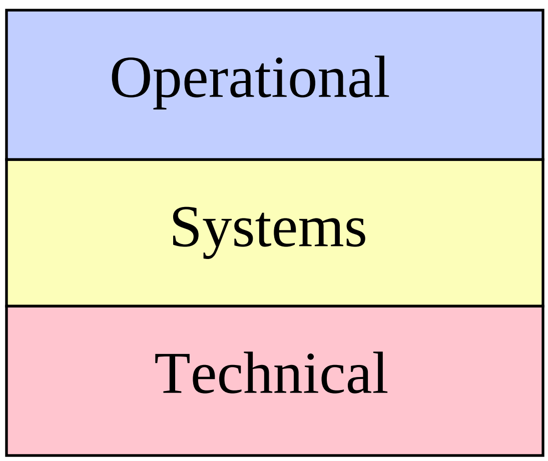
C4ISR ITF

Integrated Architectures Panel

C4ISR Architecture Framework

Version 1.0



CISA-0000-104-96

7 June 1996

Table of Contents

SECTION **PAGE**

Table of Contents (Cont.)

SECTION **PAGE**

LIST OF FIGURES

This Page Intentionally Left Blank

PREFACE

The principal objective of this effort is to define a coordinated approach, a framework, for Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) architecture development, presentation, and integration. This report represents Version 1.0 of that framework and is intended to provide a basis from which the community can work collectively to evolve the framework and promulgate it as DoD direction via appropriate DoD policy directives and guidance instructions.

The C4ISR Architecture Framework was developed under the auspices of the C4ISR Integration Task Force (ITF) Integrated Architectures Panel, whose members included representatives from the Joint Staff, the Services, the Office of the Secretary of Defense, and the Defense agencies. Direction and guidance was provided by the Architectures Directorate of the C4I Integration Support Activity (CISA), Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence (OASD(C3I)). The Framework Steering Group, composed of representatives of the Command, Control, Communications, and Computer Systems Directorate (J6) of the Joint Staff, Information Technology Directorate of OASD(C3I), Defense Information Systems Agency (DISA), and the Services, provided important insights and was instrumental in determining appropriate content.

Version 1.0 of the Architecture Framework has focused on information flow and systems-related issues. The Unified Commands frequently use architectures as a mechanism to address a broader range of operational capabilities such as those associated with people, training, facilities, management, and direction. This expanded use is a logical extension of principles discussed in the Version 1.0 report. Subsequent versions of the Framework will more specifically address this expanded use of architectures.

This Page Intentionally Left Blank

EXECUTIVE SUMMARY

PURPOSE

This report presents Version 1.0 of the Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) Architecture Framework. The Framework provides direction for developing and presenting architectures and represents a coordinated approach for C4ISR architectures in support of DoD goals and objectives. Use of the Framework by DoD organizations will ensure a common denominator for understanding, comparing, contrasting, and integrating architectures.

The Framework has been developed in close coordination with the ongoing work of the Integrated Architectures Panel of the C4ISR Integration Task Force (ITF). The Framework will be provided to the C4ISR ITF as part of the Integrated Architectures Panel's Final Report and will be proposed to the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence as formal direction to DoD. The Framework is expected to be promulgated as direction via appropriate DoD policy directives and guidance instructions.

OBJECTIVE AND SCOPE

The objective of the Framework task is to define a common approach for the military Services, unified commands, and Defense agencies to follow in developing their various C4ISR architectures. As defined in this report, the Framework provides guidelines and defines a process that can be used across DoD for developing C4ISR architectures with a focus on support to the warfighter. Although developed as a means for describing C4ISR operational, systems, and technical architectures to support warfighting tasks, the Framework can be readily extended to applicability to other architectures within the DoD such as for personnel, accounting, acquisition, etc.

The primary existing DoD guidance specifically focused on architectures is the Technical Architecture Framework for Information Management (TAFIM), developed by the Defense Information Systems Agency (DISA). The TAFIM focuses on technical architectures and establishes a framework for Standards Based Architecture (SBA) planning and information technology standards. The Framework defined in this report primarily focuses on operational and systems architectures and is intended to be complementary to the TAFIM.

CURRENT ARCHITECTURE ENVIRONMENT

There is currently no commonly used approach for architecture development and use within the DoD. The commanders in chief (CINCs), the military Services, and the DoD agencies (C/S/As) are increasingly developing and using architectures to support a variety of objectives, such as visualizing and defining operational and technical concepts, identifying operational requirements, assessing areas for process improvement, guiding systems development and implementation, and improving interoperability. Many different constructs are used to develop and portray architectures.

There are excellent initiatives, such as Air Force *Horizon*, Army *Enterprise*, and Navy *Copernicus...Forward*, but generally they are not connected. At the DoD-level, there are various architecture forums, such as the Architecture Methodology Working Group, the Architecture and Integration Council, and the Intelligence Systems Board, but they are not readily coupled.

The TAFIM's SBA Methodology describes a process to develop and achieve an integrated information technology architecture that some DoD organizations have chosen to use. Likewise, several of the Services, and some of the commands and agencies have established processes for developing, presenting, and managing architectures. The processes vary according to the organization, and some are more mature than others. This multitrack approach to the command, control, communications, computers, and intelligence (C4I) architectural world often yields stovepiped, inconsistent C4I architectures. The community is unable to fully leverage across various C/S/A architectures to develop a seamless, integrated C4ISR environment.

RESULTS OF RECENT DoD ARCHITECTURE INITIATIVES

In October 1995, the Deputy Secretary of Defense directed that a DoD-wide effort be undertaken to define and develop better means and processes for ensuring that C4I capabilities meet the needs of warfighters. To accomplish this goal, the C4ISR ITF, under the direction of the ASD (C3I), was established. This task force, consisting of representatives from the Joint Chiefs of Staff, the military Services, and DoD agencies, was organized into various panels, each charged with tackling a different aspect of the problem.

The Integrated Architectures Panel has focused on the processes for DoD to develop coherent integrated C4ISR architectures. Prior to the establishment of this ITF, differing views of architecture persisted in the DoD community at large. These differing views included functional, operational, information, physical, systems, and technical. The four Services had already decided to focus on an architecture construct consisting of operational, systems, and technical architectures. Early in the Panel's deliberation, this construct was accepted as the set of architectures required in the DoD. Through efforts of the Panel, consensus regarding the nature and roles of operational, system, and technical architectures has emerged.

The Integrated Architectures Panel has recognized the need for a common approach for developing and presenting architectures. The processes defined by the C/S/As provide a substantial foundation that can be used as a springboard to a coordinated DoD approach.

In support of the C4ISR ITF Integrated Architectures Panel, the C4I Integration Support Activity (CISA) undertook the task of leading an effort to develop a C4ISR Architecture Framework that establishes a standardized set of rules and guidance for the Services, commands, and DoD agencies to use in their development and documentation of C4ISR architectures. This Framework builds upon the results of other architecture efforts within the DoD by leveraging concepts and ideas from various efforts in such a way that they can be used together.

Once adopted, the Framework will provide a common basis for developing architectures that can be universally understood and readily compared and contrasted to other architectures. It will facilitate the reuse of architectural information and results and will serve as the foundation for expansion and integration of architectures across organizational and functional boundaries. In addition, the C4ISR Architecture Framework will promote effective communications between warfighters and system developers by providing a context within which operational analysis and systems engineering can be integrated to provide logical connectivity from strategic objectives down to processes and supporting system elements. Ultimate potentials include facilitating the creation of a joint, integrated C4ISR environment, facilitating development of common solutions for similar needs across C/S/As, and improving compatibility, interoperability, and integration among C4ISR capabilities.

Development of the Framework has been an evolutionary process, paralleling and keeping pace with the deliberations and conclusions of the Integrated Architectures Panel and its subpanels. Consequently, the Framework as presented here represents a snapshot in time of the current thinking and will evolve and be refined as it is put into practice.

ARCHITECTURE DEFINITIONS AND LINKAGES

The Integrated Architectures Panel has agreed to the following definitions:

Architecture. The structure of components, their relationships, and the principles and guidelines governing their design and evolution over time. (IEEE STD 610.12)

Operational Architecture. Descriptions of the tasks, operational elements, and information flows required to accomplish or support a warfighting function

Systems Architecture. Descriptions, including graphics, of systems and interconnections providing for or supporting warfighting functions

Technical Architecture. A minimal set of rules governing the arrangement, interaction, and interdependence of the parts or elements whose purpose is to ensure that a conformant system satisfies a specified set of requirements

These definitions clarify the distinctions among the types of architectures, emphasizing the precept that operational architectures present the functional or logical requirements for C4ISR support to the warfighter, while system and technical architectures describe the physical capabilities and attributes that actually meet operational needs. Expanded definitions of the three architecture types are provided in Section 3.

The three types of architectures can be considered as components of the architecture of a given subject area. Therefore, it is important to recognize the linkages among those components, illustrated in **Figure ES-1** below.

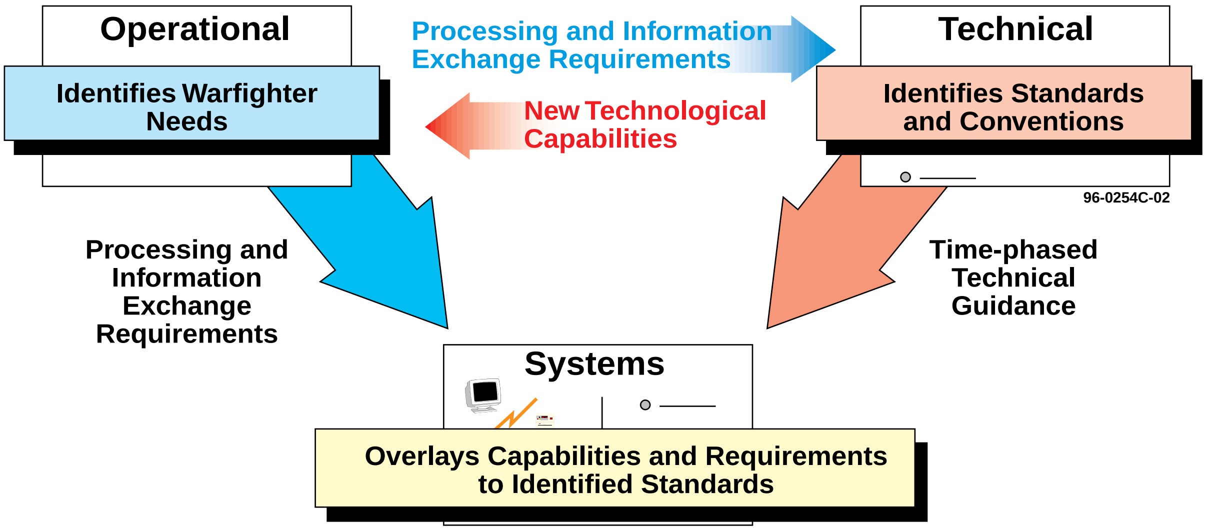


Figure ES-1: Linkages Among Architecture Types

THE FRAMEWORK FOR C4ISR ARCHITECTURES

The architecture definitions and linkages agreed to by the Integrated Architectures Panel are the foundation of the Framework. In addition, the Framework is intended to be consistent with the objectives, concepts, and methodologies contained in the TAFIM.

The Framework provides a base methodology for developing architectures. It is not intended to be a rigid "cookie-cutter" approach to architecture development. It does not mandate specific techniques or automated tools. Instead, the guidelines are intended to allow sufficient flexibility so that they do not restrict organizations in achieving their own analysis needs.

The Framework calls for consistent summary information, essential data and specific products according to architecture type. The summary information, applicable to all architectures, consists of a clear identification of the type of architecture, its scope, the purpose and intended users, the context for which the architecture is designed, and, if applicable, any findings derived from the architecture. For each architecture type, a minimum set of data is defined that must be present to satisfy the definition of that architecture type and to provide the appropriate basis for generating products.

Architecture Products

As used here, "architecture products" are graphical, textual, and tabular items that are developed in the course of building an architecture and describe characteristics pertinent to the architecture and its purpose. The set of architecture products varies depending upon the type of architecture being developed and the specific objectives and scope of the architecture. The decision of which architecture products to build is based on the issue areas the architecture is intended to explore and the resulting characteristics that the architecture must capture and describe. A given architecture may consist of all the products described in the Framework or may be a selected subset of those products. Templates are provided for developing textual and graphic architectural products based on essential information. The relationship among the architecture products is described to facilitate traceability of C4ISR solutions back to the operational warfighting and warfighter support requirements that they are aimed at satisfying.

The following are proposed standard architecture products. Each is addressed in the main body of this report with a discussion of the characteristics the product should capture; a description of how the product can be used or why it is needed; a generic, graphic template, where appropriate; and a real-world example product.

- Operational Architecture
  - Operational Concept Diagram
  - Command Relationship Chart
  - Activity Model
  - Information Exchange Requirements
  - Required Capabilities Matrix
  - Basic Node Connectivity Model
- Systems Architecture
  - Systems Overlays (to Basic Node Connectivity Model)
  - System Element/Interface Diagram
  - System Performance Parameters Matrix
  - System Evolution Diagram
- Technical Architecture
  - Tailored Technical Criteria Profile
  - Technology Forecast
- Core Products
  - Data Model
  - Data Dictionary

The data dictionary, which contains definitions of all significant terms used in the other products, is not assigned to any one architecture type because it supports and is populated by all three. Similarly the data model is not assigned to any one type of architecture.

SUMMARY

The principal conclusion arising from this initial effort is that common terms of reference, common definitions, and a common Framework for documenting architectures will significantly improve DoD's ability to achieve a seamless, integrated C4ISR environment. The Framework presented here is an initial step towards achieving commonality and will evolve over time as the Services, DoD agencies, and commands apply it to improve C4ISR support to the warfighter.

SECTION 1
INTRODUCTION

1.1 PURPOSE

This report presents Version 1.0 of the Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework for the development and presentation of architectures and represents a first attempt to define a coordinated approach for C4ISR architecture in support of DoD goals and objectives. The Framework provides the rules and guidance for developing and presenting architectures that ensure a common denominator for understanding and comparing architectures.

1.2 OBJECTIVE AND SCOPE

As implied by the report title, the scope of this initial step in framework development is directed at command and control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) architectures with the focus on C4ISR support to the warfighter. The objective was to develop a common unifying approach for the military Services, unified commands, and Defense agencies to follow in developing their various architectures. While the specific focus has been C4ISR, the approach defined in the framework is readily extendible to other DoD functional areas such as personnel management, systems acquisition, and finance.

1.3 ORGANIZATION OF THIS DOCUMENT

The remainder of this document is organized as follows:

Section 2, Background, describes the current architecture environment within the DoD that led to development of the Framework, recent architecture initiatives related to the Framework, and the process through which this Framework has evolved.

Section 3, Overview of the Framework, presents the analytical basis for the Framework in terms of its relationship to prior and other ongoing efforts, the relationship between military doctrine and architectures, and the linkage between architectures to warfighter missions, tasks and activities, and the information needed to support them. It describes operational, systems, and technical architectures, the relationship among them, and the types of architecture information needed to establish those linkages. The Framework also describes the necessary structure for documenting architectures to facilitate their understanding, comparison, and integration; and provides guidance on how the resulting architectural documentation may be used.

Section 4, Product Descriptions, presents descriptions of the architecture products and provides guidance for their development and use.

This Page Intentionally Left Blank

SECTION 2

BACKGROUND

2.1 CURRENT ARCHITECTURE ENVIRONMENT

There is currently no common approach for architecture development and use within the Department of Defense (DoD). The unified commanders in chief (CINCs), the military Services, and the DoD agencies (C/S/As) are increasingly developing and using architectures to support a variety of objectives, such as visualizing and defining operational and technical concepts, identifying operational requirements, assessing areas for process improvement, guiding systems development and implementation, and improving interoperability. There are many different views and approaches in use.

One factor contributing to the variation in approaches is the diversity of purposes that the architecture serves, such as:

- Developing joint requirements for program mission need statements (MNSs) and operational requirements documents (ORDs)
- Identifying and prioritizing C4ISR system deficiencies and allocations in context with joint needs
- Improving interoperability and identifying opportunities for integration
- Determining policy/doctrine, system support needs, and application/infrastructure support needs for a specific joint warfighting functions
- Identifying communications connectivity and capacity requirements
- Measuring system strengths and weaknesses with respect to supporting joint operations

Consequently, there has been no common agreement within DoD concerning what architectures are and what they can or should be able to do.

The current C4ISR environment in the DoD does not reflect the existence of an institutionalized process for architecture development. The Technical Architecture Framework for Information Management's (TAFIM's) Standards Based Architecture (SBA) Methodology describes a process to develop and achieve an integrated information technology architecture that some DoD organizations have chosen to use. All of the Services and some of the commands and agencies have established processes for developing, presenting, and managing architectures. The processes vary according to the organization, and some are more mature than others. Architectures are frequently initiated on an ad hoc basis when some authority directs action. The action is constrained by diverse local guidance that varies according to organization, is facilitated by disparate tools, and culminates in some form of descriptive documentation or assessment. The highly decentralized processes in place have led to inefficiencies in architecture development, such as collecting and cataloging information that cannot be readily reused or failing to focus on specific critical aspects of architecture development such as a clear understanding of the warfighting functions to be supported.

The result has been a widespread perception that architectures are stovepiped, piecemeal, and disjointed. This has lead to a general lack of confidence in architectures throughout the DoD that has often culminated in reliance on “squeaky wheels” as the means to identify problems and the use of “gut feel” to develop solutions.

Several attempts have been made by DoD and the Services to remedy this situation. Examples include the Navy’s *Copernicus...Forward*, the Army’s *Enterprise*, and the Air Force’s *Horizon* strategies for developing integrated C4I architectures, and Defense Information Systems Agency’s (DISA’s) Architecture Methodology Working Group (AMWG), Automated Architecture Tool Suite (AATS), and Standard Data Element-Based Automated Architecture Support Environment (SAASE). Their efforts represent a strong start, but lack the centralized top-level support needed to develop a unified DoD-wide strategy for ensuring C4ISR capabilities meet warfighters’ needs.

2.2 RESULTS OF RECENT DoD ARCHITECTURE INITIATIVES

In October 1995, the Deputy Secretary of Defense directed that a DoD-wide effort be undertaken to define and develop better means and processes for ensuring that C4I capabilities meet the needs of warfighters. To accomplish this goal, the C4ISR Integration Task Force (ITF) was established under the direction of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence (ASD [C3I]). This task force, consisting of representatives from the Joint Chiefs of Staff, the military Services, and DoD agencies was organized into sets of panels and subpanels, each charged with tackling a different aspect of the problem.

The Integrated Architectures Panel has focused on the processes for DoD to develop coherent integrated C4ISR architectures. Prior to the establishment of this ITF, a variety of differing views of architecture persisted in the DoD community at large. These differing views included functional, operational, information, physical, systems, and technical. The four Services had decided to focus on an architecture construct consisting of operational, systems, and technical architectures. Early in the Panel’s deliberation, this construct was accepted as the set of architectures required in the DoD. The Panel has produced a set of agreed definitions of Operational Architecture, Systems Architecture, and Technical Architecture. Also, a common understanding of the nature and roles of operational, system, and technical architectures has emerged.

2.3 EVOLUTION OF THE C4ISR ARCHITECTURE FRAMEWORK

In support of the C4ISR ITF Integrated Architectures Panel, the C4I Integration Support Activity (CISA) undertook the task of leading an effort to develop a Framework for C4ISR architecture development that establishes a standardized set of rules and guidance for the Services, commands, and DoD agencies to use in the development and documentation of C4ISR architectures. This Framework builds upon other architecture efforts within the DoD, as shown in **Figure 2-1**, by combining many of the concepts and ideas from them so they can be used together.

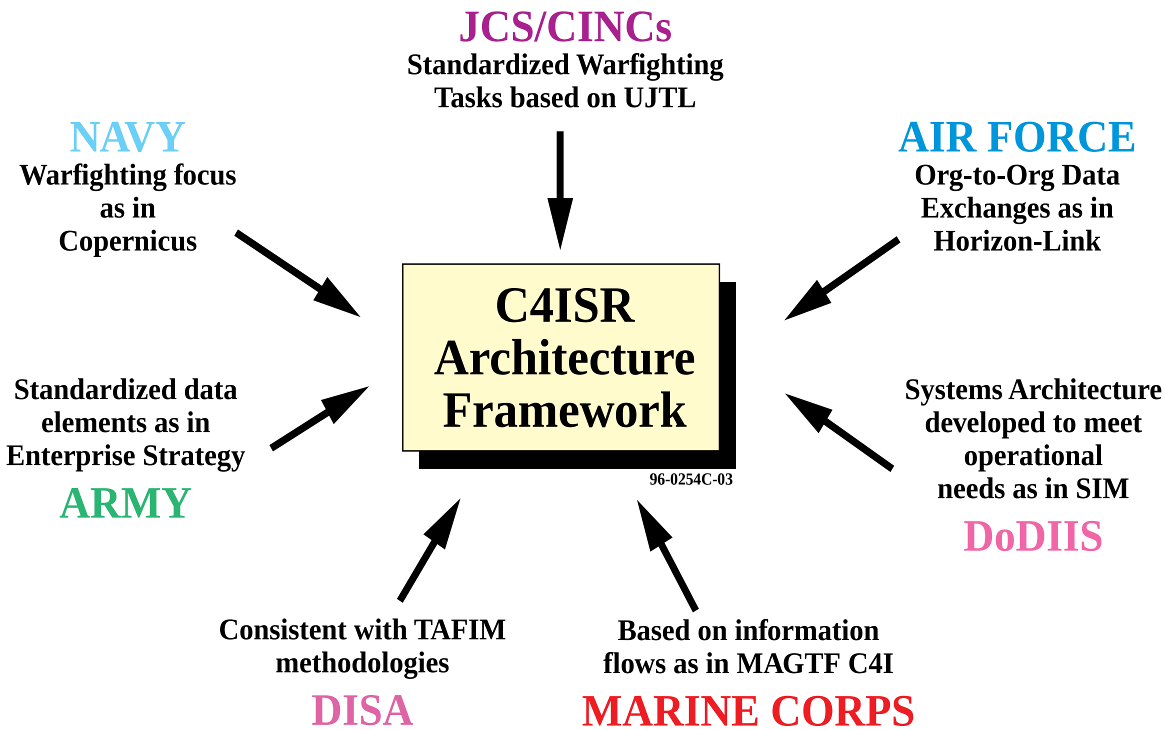


Figure 2-1: Leveraging Prior Efforts

Once adopted, the architecture Framework will provide a common basis for developing architectures that can be universally understood and readily compared to other architectures, will facilitate the reuse of architectural information and results, and will serve as the foundation for expansion and integration of architectures across organizational and functional boundaries. In addition, the C4ISR Architecture Framework will promote effective communications between warfighters and system developers by providing a context within which operational analysis and systems engineering can be integrated to provide logical connectivity from strategic objectives down to processes and supporting system elements. Ultimate potentials include:

- Ensuring that DoD is acquiring capabilities that focus on the evolving needs for joint/combined operations
- Creating a joint, integrated C4ISR environment capable of effective multinational operations
- Encouraging, facilitating, and emphasizing development of common solutions for similar needs across CINCs, Services, and DoD agencies

- Facilitating, improving, and ensuring compatibility, interoperability, and integration among C4ISR capabilities
- Facilitating, encouraging, and ensuring the fielding of joint, integrated, and interoperable C4ISR capabilities that meet operational and support needs

Development of the Framework has been an evolutionary process, paralleling and keeping pace with the deliberations and conclusions of the Integrated Architectures Panel and its subpanels. Consequently, the framework as presented here represents only a snapshot in time of the current thinking, and will evolve and be refined as it is put into practice.

SECTION 3

OVERVIEW OF THE FRAMEWORK

3.1 ANALYTICAL BASIS OF THE FRAMEWORK

The proposed Framework is based on the concept of operational, systems, and technical architectural components that work in concert. Although initially developed as a means for describing C4ISR operational, systems, and technical architectures to support warfighting tasks, the Framework that has been developed can be readily extended to other information architectures within the DoD such as for personnel management, systems acquisition, or finance.

3.1.1 Relationship to DoD Policy and Guidance Documents

The primary existing guidance on architectures is the TAFIM; DoD Directive (DoDD) 5105.19, and Chairman of the Joint Chiefs of Staff (CJCS) Memorandum of Policy (MOP) 50 and its proposed replacement CJCS Instruction (CJCSI) 6111.11. The TAFIM focuses on technical architectures and establishes a framework for SBA planning and information technology standards. DoDD 5105.19 tasks DISA with developing and maintaining architectures to ensure end-to-end interoperability of strategic and tactical C4 and information systems used by the National Command Authorities and DoD components. CJCS MOP 50/Instruction 6111.11 establishes policy and guidance for the development and assessment of C4 architectures and planning documents. However, there is no DoD policy, directive, or other guidance for developing and maintaining operational and systems architectures.

The proposed C4ISR Architecture Framework is consistent with the objectives, concepts, and methodologies contained in the TAFIM and provides the necessary extension of these concepts to the development of operational and systems architectures.

3.1.2 Architecture Definitions

The architecture definitions agreed to by the Integrated Architecture Panel provide the foundation for the Framework.

Architecture. The structure of components, their relationships, and the principles and guidelines governing their design and evolution over time. (IEEE STD 610.12)

Operational Architecture. Descriptions of the tasks, operational elements, and information flows required to accomplish or support a warfighting function

Systems Architecture. Descriptions, including graphics, of systems and interconnections providing for or supporting warfighting functions

Technical Architecture. A minimal set of rules governing the arrangement, interaction, and interdependence of the parts or elements whose purpose is to ensure that a conformant system satisfies a specified set of requirements

These definitions clarify the distinctions among the types of architectures, emphasizing the precept that operational architectures present the functional or logical requirements for C4ISR support to the warfighter, while system and technical architectures describe the physical capabilities and attributes that actually meet operational needs.

3.1.3 Guiding Principles

There were several guiding principles that drove the development of the Framework. These are that architectures should:

- Be built with a purpose
- Facilitate user understanding and communication among users
- Permit comparison and integration
- Be modular, expandable, and reusable

First, development of architectures should be driven by a particular purpose, and that purpose must be clearly understood. Architectures may be developed to provide an understanding of the complex relationships among tasks, operational elements, information flows, and systems. Within C4ISR, the range of systems to be addressed extends from the sensor, through processing and information systems to the shooter, to include associated communications. Architectures may also be developed to address more focused objectives such as identifying process improvement opportunities, defining and evaluating particular capabilities such as communications capabilities, or examining actual versus required interoperability in a subject area. A clear identification of the purpose will provide the proper perspective for selecting the type of architecture to be built, the depth and breadth that it should cover, and the particular way its information should be portrayed. Since architectures may serve multiple purposes, the various purposes the architecture may ultimately serve should be explored in advance of architecture development so the most efficient use may be made of the often time-consuming data-gathering steps that precede the actual development process.

To facilitate user understanding and communication among users, the ways of presenting architecture information should be chosen with the background and perspective of the expected users in mind. Such methods may include graphics, tables, databases, and text. In addition, the level of detail presented must be tailored to the particular needs of the user and the purposes that the architecture may serve.

The key prerequisite to both the comparison and integration of architectures and to modularity, expandability, and reusability is to build architectures using commonly understood terms and approaches. Definition of a set of essential architecture information, like plastic children's building blocks that come in standard sizes, shapes, and colors, will permit architectures to be built modularly, integrated with each other, expanded upon, and reused. As used here, "integratable" does not necessarily imply that one architecture should be able to "plug into" another or that multiple architectures be physically combined into one architecture document. In this context integration means to analyze multiple architectures to serve a common purpose to identify linkages, gaps, overlays, and redundancies. Even if built for different purposes, architectures built using similar terms and methods will facilitate comparison and make them easier to adapt for other uses.

There are several types or levels of integration: one is integration between hierarchical levels, such as between joint-level and Service- or lower organization-level architectures; another is integration between different architectures at the same hierarchical level; still another is integration among Operational, Systems, and Technical Architecture Components within a single architecture. **Figure 3-1** illustrates these various types of integration. For example, a Navy

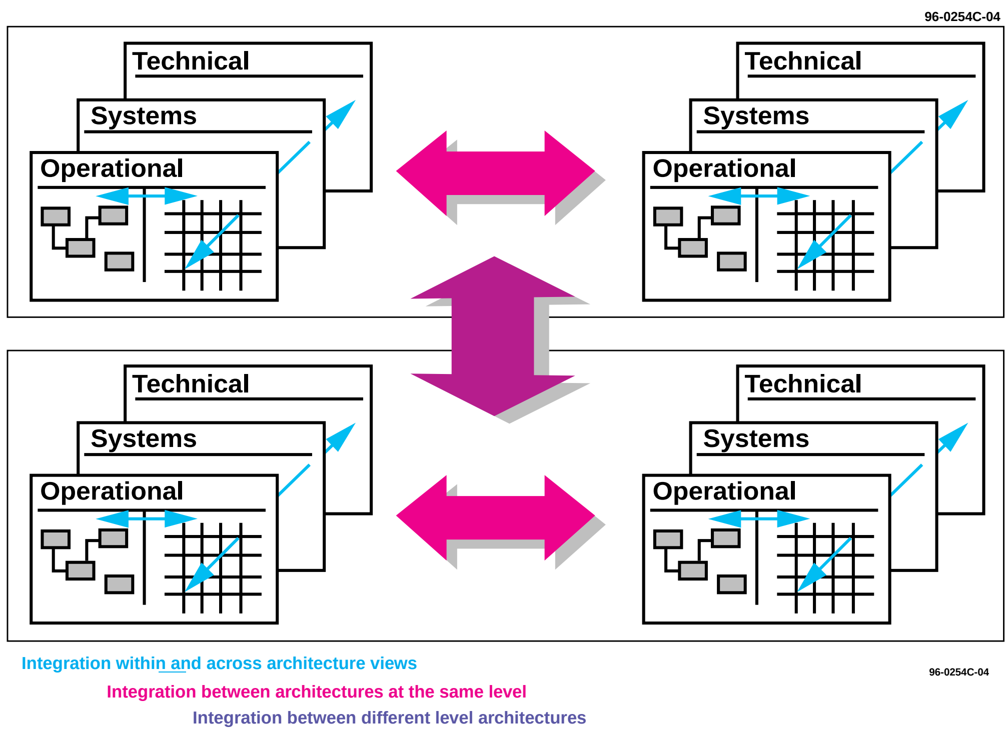


Figure 3-1: Architecture Integration

architecture on Mine Warfare could be related to a Joint Maritime architecture. An operational architecture for precision strike could be compared to one for integrated air defense to identify where the two may be able to take advantage of common information sources. Alternatively, a systems architecture for theater missile defense may be compared to one for theater air defense to determine if there are opportunities for sharing communications systems and processing algorithms. Within the same subject area, operational and systems architectures for mine warfare may be compared and linked to each other to ensure that the right systems are in place to meet operational needs.

The end result will be broader understanding of architectures, reduction in the time and expense to build architectures, and ultimately the capability to develop overarching architectures that cut across multiple organizational and functional bounds.

3.1.4 Relationship to Military Doctrine

The Framework uses existing military doctrine as the basis for defining the missions, tasks, and activities upon which architectures are based. However, the Framework also recognizes the fact that military doctrine is never static, that it is constantly evolving in response to changes in military strategy and opportunities to exploit advances in technology. This may be particularly true of the rapid revolution in technology associated with C4ISR capabilities.

3.2 OPERATIONAL, SYSTEMS, AND TECHNICAL ARCHITECTURES

As a result of the deliberations of the Integrated Architecture Panel, there is now a better understanding within the DoD of the nature and roles of operational, systems, and technical architectures. The key distinctions among these architectures and the relationships among them are depicted in **Figure 3-2**. As shown in this figure, operational architectures are used to identify

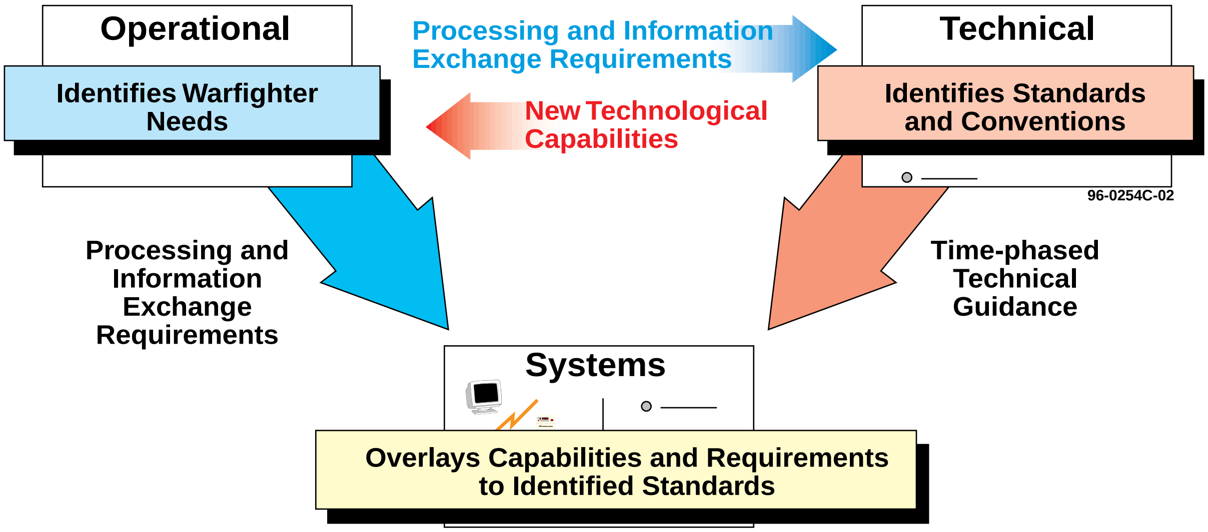


Figure 3-2: Linkages Among Architecture Types

and document warfighter C4ISR needs,* systems architectures are used to describe how the needs can be met using identified standards and conventions, and technical architectures specify

* It should be noted that the formal establishment and validation of operational requirements is accomplished via the existing Mission Need Statement (MNS) and Operational Requirements Document (ORD) process.

the standards to be used for meeting the needs. In addition, the operational architectures provide the context for exploration and time-phased selection of new technologies, while the technical architectures provide new technologies that can be exploited through new operational concepts.

The Operational Architecture, Systems Architecture, Technical Architecture and Integrated Subpanels of the Integrated Architecture Panel have expanded the basic definitions of architecture types and have defined basic tenets to which each architecture type adheres. The expanded definitions and tenets are provided in **Figures 3-3, 3-4,** and **3-5**.

Operational Architecture

Expanded Definition

A description (often graphical) of the operational elements, assigned tasks, and information flows required to support the warfighter. It defines the type of information, the frequency of exchange, and what tasks are supported by these information exchanges.

Tenets

- **Primary Purpose of an Operational Architecture:**
  - Defines activities and information exchange requirements
- **Operational Architecture Characteristics**
  - Operational architectures start with doctrine and assigned tasks, which in turn drive the definition of the activity model.
  - Activity descriptions are essential for defining the data model and information exchange requirements.
  - Information exchange requirements may cross organizational boundaries.
  - Activities may cross organizational boundaries.
  - Operational architectures should not be systems-dependent.
  - Activity descriptions are not based on organizational models or force structure.

Figure 3-3: Operational Architecture Expanded Definition and Tenets

Systems Architecture

Expanded Definition

Defines the physical connection, location, and identification of key nodes, circuits, networks, warfighting platforms, etc., and specifies system and component performance parameters. The systems architecture is constructed to satisfy operational architecture requirements per standards defined in the technical architecture. The systems architecture shows how multiple sysems within a subject area link and interoperate, and may describe the internal construction or operations of particular systems within the architecture.

Tenets

- **Primary Purpose of an Systems Architecture:**
  - Enables or automates operational activities through physical processes.
- **Systems Architecture Characteristics**
  - Operational architectures drive associated systems architectures.
  - Systems architectures map systems with their associated platforms, functions, characteristics, and data elements back to the operational architecture.
  - Systems architectures identify system interfaces and define the connectivity between systems.
  - Systems architectures define system constraints and bounds of system performance behavior.
  - The systems architecture shows systems interconnectivity from sensor-to-shooter/decisionmaker through the system components.
  - Systems architectures are technology-dependent, show how multiple systems within a subject area link and interoperate, and may describe the internals of particular systems.
  - Systems architectures support multiple command organizations and missions; the clustering of system functions and data stores shown in the systems architecture should not be based on current organizational models, force structures, or fielded technologies.

Figure 3-4: Systems Architecture Expanded Definition and Tenets

Technical Architecture

Expanded Definition

The technical architecture identifies the services, interfaces, standards, and their relationships. It provides the technical guidelines for implementation of systems upon which engineering specifications are based, common building blocks are built, and product lines are developed.

Tenets

- **Primary Purpose of a Technical Architecture:**
  - Defines the set of rules that govern systems implementation and operation.
- **Technical Architecture Characteristics**
  - Technical architectures are based on requirements defined in the operational architecture and analyses of possible enabling technologies.
  - Information system paradigms of processing, database and communications are identified and strongly influence the technical architecture.
  - Technical architectures account for the requirements of multiplatform and network interconnections among all systems that produce, use, or exchange information electronically.
  - Definitions and corresponding technical criteria for system capabilities, services, and interfaces are provided in the technical architecture.
  - Technical architectures accommodate new technology, evolving standards, and the phasing out of old technology.
  - The rules of technical architectures are defined in terms of nonproprietary specifications and therefore reduce reliance on proprietary technologies.

Figure 3-5: Technical Architecture Expanded Definition and Tenets

Architectures are developed according to a defined scope and within a specific context. The scope includes the architecture type, subject area, and time frame for which the architecture is applicable. In general, the subject area for operational architectures is based upon mission areas such as Joint Maritime Operations, Mine Warfare, and Theater Air Defense. The interrelated conditions that compose the setting in which the architecture exists constitute the context for the architecture. The context includes such things as doctrine; tactics, techniques, and procedures; relevant goals and vision statements; and concepts of operations, scenarios, and environmental conditions. High-level, broad scope architectures embrace the range of potential physical, military, and civil environmental conditions so that the resulting architectures are highly stable and are relatively insensitive to moderate changes in environmental conditions. Specific environmental conditions are reflected in operational plans and may also be more directly reflected in lower level, issue-focused architectures.

In the context of C4ISR architectures, system architectures are expected to address the full range of automated systems from sensors that collect information and pass it on, through processing and information systems, communications systems, and shooters that require automated information to accomplish their objectives. System architectures depict the functional and physical automated systems, nodes, platforms, communications paths, and other critical elements that provide for supporting information exchange requirements and warfighter tasks described in the operational architectures. Various attributes of the systems, nodes, and required information exchanges are included according to the purpose of the specific architecture effort.

Tactical, strategic, and support systems must be able to "plug and play" in a joint, global environment. To achieve this ability, there must be a mechanism for incorporating information technology consistently, controlling the configuration of technical components, and ensuring compliance with technical "building codes." The technical architecture provides this mechanism.

Technical architectures are intended to transition the logical operational architecture to the physical systems architecture by providing the fundamental building blocks on which to base development. Well-planned and comprehensive technical architectures facilitate integration and promote interoperability across systems and compatibility among related architectures. As part of a disciplined process to build systems, technical architectures reduce information technology costs across an organization by highlighting risks, identifying technical or programmatic issues, and driving technology reuse. Adherence to a technical architecture streamlines and accelerates systems definition, approval, and implementation.

3.3 DOCUMENTATION OF ARCHITECTURES

Architectures should be developed via a common approach that includes providing summary information, a minimum set of essential architecture information, and specific architecture products. The summary information, applicable to all architectures, consists of a clear identification of the type of architecture, its applicable time frame, and the purpose and intended users of the architecture. As used here, the term "architecture products" refers to graphical, textual, and tabular items that are developed in the course of building an architecture. Architecture products describe characteristics pertinent to the architecture and its intended purpose. The set of architecture products used in a given architecture effort vary depending upon the type of architecture being developed, the scope, and the specific purpose and objectives of the architecture. The framework provides templates for development of these textual and graphic architectural products using consistent terminology and display techniques to facilitate common understanding and to provide a basis for comparing, and integrating architectures when it is beneficial to do so.

3.3.1 Common Summary Information

To facilitate understanding and provide opportunities for comparing and integrating architectures, all architecture documentation should include the information described below. This information should be provided as part of an executive summary or overview section at the beginning of the architecture documentation.

- **Scope.** The first item that should be included in the introductory information is a clear identification of the scope of the architecture in terms of type, subject area, and temporal nature.

  Type. Operational, systems, or technical.

  Subject Area. The applicable operational, systems, or technical areas or domains for which the architecture is aimed. For example the subject area could be mine warfare in littoral areas, or precision strike. The subject area could address infrastructure such as satellite communications or UHF broadcasts. Technical architectures may cover the range of standards necessary for a given operational or systems architecture or could focus on a specific area such as data encryption standards for direct broadcast systems.

  Temporal Nature. The time frame for which the architecture is applicable. Examples of words used to express temporal nature are "current," "as is," "baseline," "to be," "target," "objective," etc.

- **Purpose and Intended Users**. Architectures should also clearly state why they have been developed and for whom.

  Purpose. Why the architecture was developed. Examples include to document existing and desired capabilities," to determine operational requirements that must be accommodated in a systems architecture, "to provide a basis for determining applicability of a new technology," and "to provide a basis for identifying the best solution to some identified problem.

  Intended Users. Who the architecture is intended to serve. Examples include a full range of potential users such as a regional CINC's J6, the JCS/J3, JTF Component Commander, the Military Intelligence Board, etc.

- **Context**. The interrelated conditions that compose the setting in which the architecture exists constitute the context for the architecture. The context includes such things as doctrine; tactics, techniques, and procedures; relevant goals and vision statements; concepts of operation; scenarios; and environmental conditions. Known or anticipated linkages to other architectures should be identified. Specific assumptions and constraints regarding the architecture development effort should be documented.

- **Findings (where applicable)**. For some architectures, particularly those aimed at providing a basis for assessments, a description of the final results of the architectural effort should be presented. Examples of results can include identification of shortfalls, recommended system implementations, opportunities for technology insertion, etc.

3.3.2 Architecture Information

Scope and context describe the basic conditions under which the architecture applies, and as such are an essential aspect of any type of architecture. Architectures should present, as required, the following information:

- Missions to be accomplished
- Operational elements to whom the missions are assigned
- Nodes where the operational elements are located
- Tasks and activities that the operational elements perform
- Information flows required to perform the tasks, to include specification of warfighter information
- Systems used to support the accomplishment of tasks and the flow of information
- System attributes and performance parameters
- The technical criteria that govern development and implementation of the systems

The level of detail and specificity to which the above are presented depends on the purpose and objectives of a given architecture effort. **Figure 3-6** (on the next page) provides a graphic representation that shows how information is associated with the three types of architectures. The set of products discussed in Section 4 includes all the essential information described above.

The set of information required for each architecture type is discussed below. Scope and environment are necessary aspects that should be specified for any architecture type. A matrix construct has been used in **Figures 3-7** (on the next page) and **3-9** (on page 3-13) to represent the essential information. It is not expected that a given architecture would present this information via a matrix. For an actual architecture, the information would likely be so extensive that presenting it in a hard copy matrix may be impractical and would probably not be useful. Instead this information readily lends itself to being stored in an automated database with only high-level summary or highlighted summary information provided in the hard copy report. Annex A presents an entity-relationship (E-R) data model showing how these sets of information are related to each other.

3.3.3 Operational Architecture Information

The essential data derives from the definition of an operational architecture (tasks, operational elements, and information flows) and attributes necessary to describe the information flow. The relationship across the three basic entities (tasks, operational elements, and information flow) is expressed in the information exchange requirement (IER). The IER is defined as the requirement for information to be passed between and among forces, organizations, or administrative structures concerning ongoing activities. IERs identify *who* exchanges *what* information with

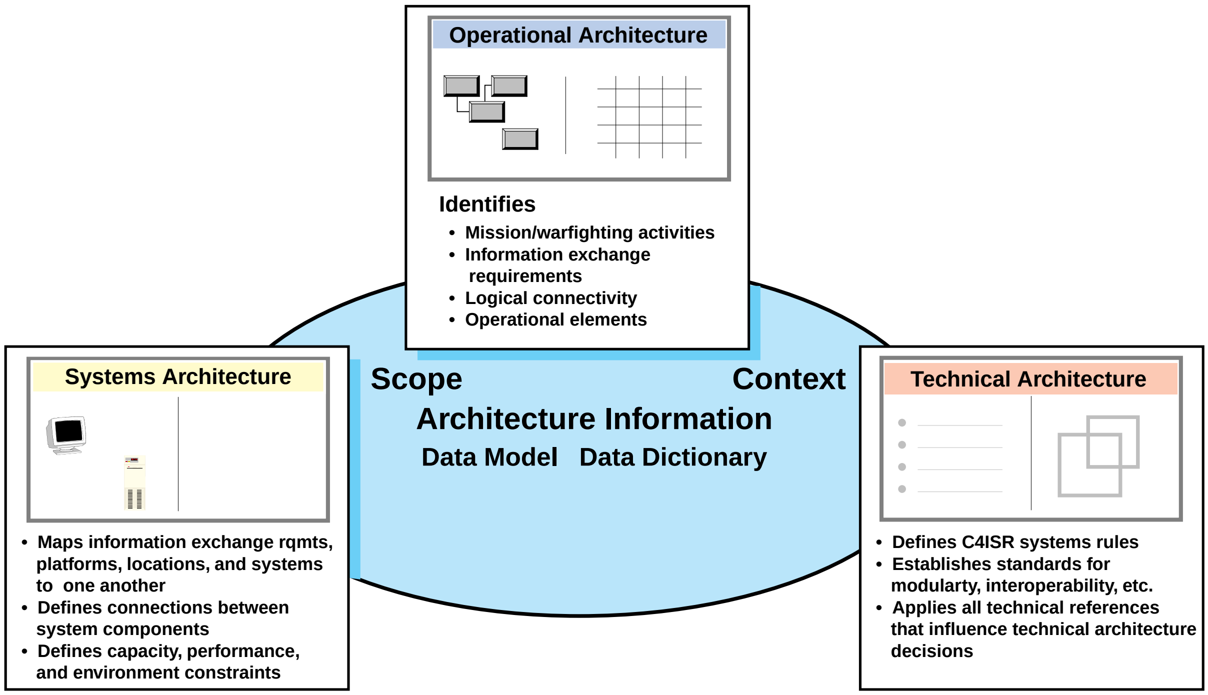


Figure 3-6: Architecture Overview

whom, as well as *why* the information is necessary and *how* that information will be used. The quality (i.e. frequency, timeliness, security), quantity (i.e. volume, speed), and type of information (i.e. data, voice, video) are attributes of the information exchange included in the IER. IERs may also identify particular capabilities needed such as large screen displays, interactive database query, color graphics, etc. The specific attributes used in any given operational architecture will depend on the purpose and level of detail of that effort.

Depending on the specific objectives of the operational architecture, information requirements may be specified in a data model with an accompanying data dictionary, using DoD standard data where possible.

Tasks should be related to the JCS's Universal Joint Task List (UJTL). **Figure 3-8** shows how the UJTL can be extended down, through derivative unified command-level Joint Mission Essential Task Lists, Service-level Mission Essential Task Lists, to the missions required to be performed by an operational element. The UJTL covers all DoD warfighting tasks such as gathering intelligence, maneuvering forces, and delivering weapons on targets as well as nonwarfighting tasks such as providing logistics support, acquiring weapon systems, and training personnel. It provides a common basis for defining any mission, task, or activity for which a C4ISR architecture must be defined.

A node is defined as a sender or receiver of information or data. In the context of operational architectures, a node could be an organization, organizational element, an activity, or even a person depending on the objectives of the specific architecture and the level at which the archi-

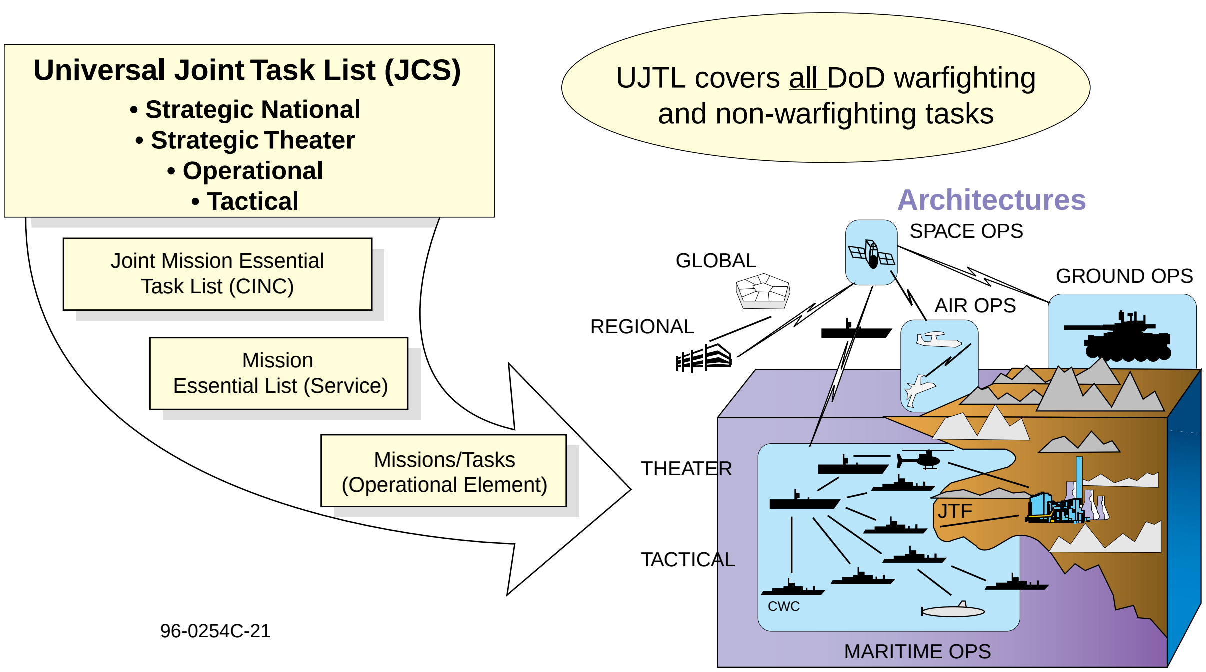


Figure 3-8: Common Basis for Defining Missions and Tasks

Information Processing Exchange	INFORMATION					SYSTEMS												INTERCONNECTIONS			
	Content	Characteristics				NODE A						NODE B						Comm Sys Name	Characteristics		
						Sys Name	Platform	Characteristics				Sys Name	Platform	Characteristics					Capacity		
@ A																					
A-B																					
......																					
@ B																					
B-A																					

Figure 3-9: Systems Architecture Essential Information

tecture was being developed. Either notional or physical assignments to nodes may be used depending on the needs and objectives of the specific architecture effort.

The information elements exchanged must be identified along with their relevant attributes. There is no single, universally accepted list of elements of information within the DoD that can provide a common basis of understanding of information flows. As an initial step to fill this gap, the Framework proposes as a strawman the definition of four major categories of warfighter information into which virtually all types of information needed to support warfighting activities can be placed. These categories are information about the enemy, friendly forces, the environment, and the situation. An initial top-level description of Elements of Warfighter Information is presented in **Annex D** as a starting point. A common list of information elements, agreed upon by the JCS, the Services, and the DoD agencies akin to the UJTL will facilitate identification of opportunities for sharing information among operational elements, establishing linkages among systems that can provide such information, and promoting the development of technical standards for information exchange.

As the framework evolves, common definitions should also be developed to describe other categories of architecture information such as required capabilities and operational elements.

3.3.4 Systems Architecture Essential Information

Systems architectures capture the information flows between nodes, the systems supporting that flow, and the communication systems supporting the interconnection. As defined above, nodes are senders, processors, and/or receivers of information. As shown in Figure 3-9, nodes are identified by the relevant system and platform. Nodes may also be described by identifying the organization and the location associated with the node. The organization and location may be physical or notional depending on the specific architecture.

Systems architectures also include information and system characteristics appropriate to the objectives of the architecture effort. It is important to note that what constitutes a node depends upon the level of detail represented in a given architecture. For example, in a high-level architecture, “U.S. Army” may be considered a node, but in a detailed architecture “workstation A” may be a node. The characteristics captured in the systems architecture will depend on the purpose of that architecture, but could include such things as applications, operating environment object names, external interface specifications, and security level. For communication interconnections, the characteristics could include such things as capacity, security level, performance bounds, electronic waveform, and transport media.

3.3.5 Technical Architecture Essential Information

The technical architecture should contain the services, configurations, standards, and convention that are to be implemented in the systems architecture. This technical guidance should be provided in a time-phased manner. As in the operational and systems architecture, scope and context must be specified for a technical architecture. The scope should include the specification of the subject area and its bounds, and also the timeframe considered in the architecture. The following are other essential elements of a technical architecture:

- **Specification of Architecture Models** - the conceptual paradigms of the processing, database, and communications parts or elements of a system required for the operating environment. Examples of these models are (1) computing: host-based and client/server; (2) database: relational, object-oriented, distributed, and centralized; and (3) communications: local area network (LAN), metropolitan area network (MAN), and wide area network (WAN). These models provide a basis for technical standards selection.
- **Specification of an Operating Environment (OE)** - the specific implementation of the appropriate common reference model, such as the Joint Technical Architecture (JTA). It defines and integrates support applications, platform services, and interfaces that constitute the infrastructure. It also provides a basis for technical standards selection. A technical architecture is a set of “building codes” for a collection of “building blocks” that must be viewed as a single entity. An OE, by defining how the “building blocks” interact and fit together, is the core of that single entity. Individual OE segments often provide a complete service or a part of a service to other segments in addition to the external environment. Thus each element of the OE cannot be considered as an independent function but must be considered in consonance with the other elements. As the core, the OE defines a set of services and interfaces common to all systems at and below that level.

- **Specification of Standards** - the complete, consistent suite of guideline documentation that reflects consensus among the affected organizational bodies on products, practices, or operations. Based upon the OE and architecture models, it provides a profile of technical standards and descriptions of standards deficiencies. The profile must specify the features, options, and extensions of each standard to the level of detail necessary to meet the TA's objectives.
- **Data Dictionary** - a repository of information about data such as definition, relationships to other data, origin, usage, and format. It provides a common reference point for the definition of information, documents information structures, and identifies standard data elements. Standard data elements are the fundamental feature enabling interoperability of all systems; they are the basis for common interpretation, processing, display of information, and communications efficiency. While the data dictionary is an essential aspect of a technical architecture, it is also important for operational and systems architectures.
- **Technical References** - the set of references such as policy, directives, conventions, transition guidance, emerging technologies, compliance criteria, and common practices that influence architectural decisions.

3.3.6 Architecture Information Relationships

As illustrated in **Figure 3-10** (on the next page), from the operational architecture the IERs, which define the required flow of information between nodes in support of a task, map into the portion of the systems architecture that defines the information being passed from node to node. Through this mapping the IERs are associated with the systems supporting the information. Similarly, the systems with their associated platforms and characteristics within the systems architecture can be associated back to the information flow, operational elements (nodes), and operational activities in the operational architecture. The standards defined in the technical architecture map into the portion of the systems architecture that describes the system characteristics.

3.3.7 Architecture Products

A common set of architecture products should be used to present architectural information in a consistent way. **Figure 3-11** (on the next page) presents the initial candidate set of architecture products. The products are presented starting from the operational architecture products upon which the systems products are based and ending with the technical architecture products. Some products, such as the Systems Overlays on the Node Connectivity Diagrams, can be viewed as being either operational or system architecture products, or both. This overlap is due to differences in perspective and in the particular needs of architecture developers and users. What matters is to capture the information that these particular products call for and not what type of architectural product they represent. Data models and a data dictionary are considered core products because they are relevant to each architecture type.

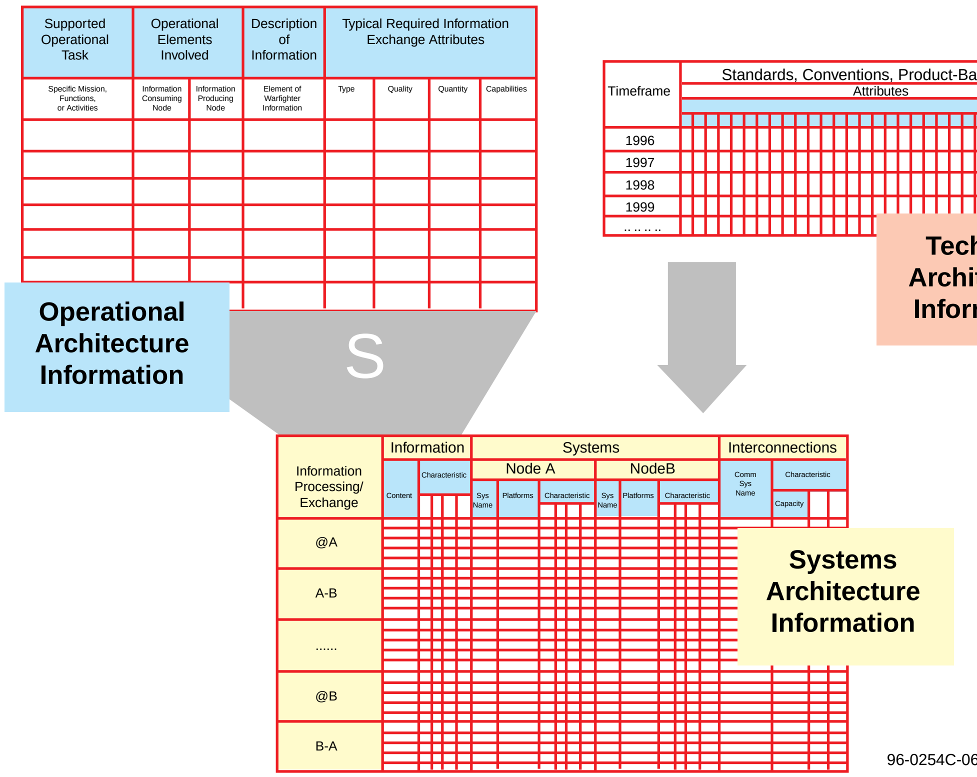


Figure 3-10: Linkages Between Architecture Types

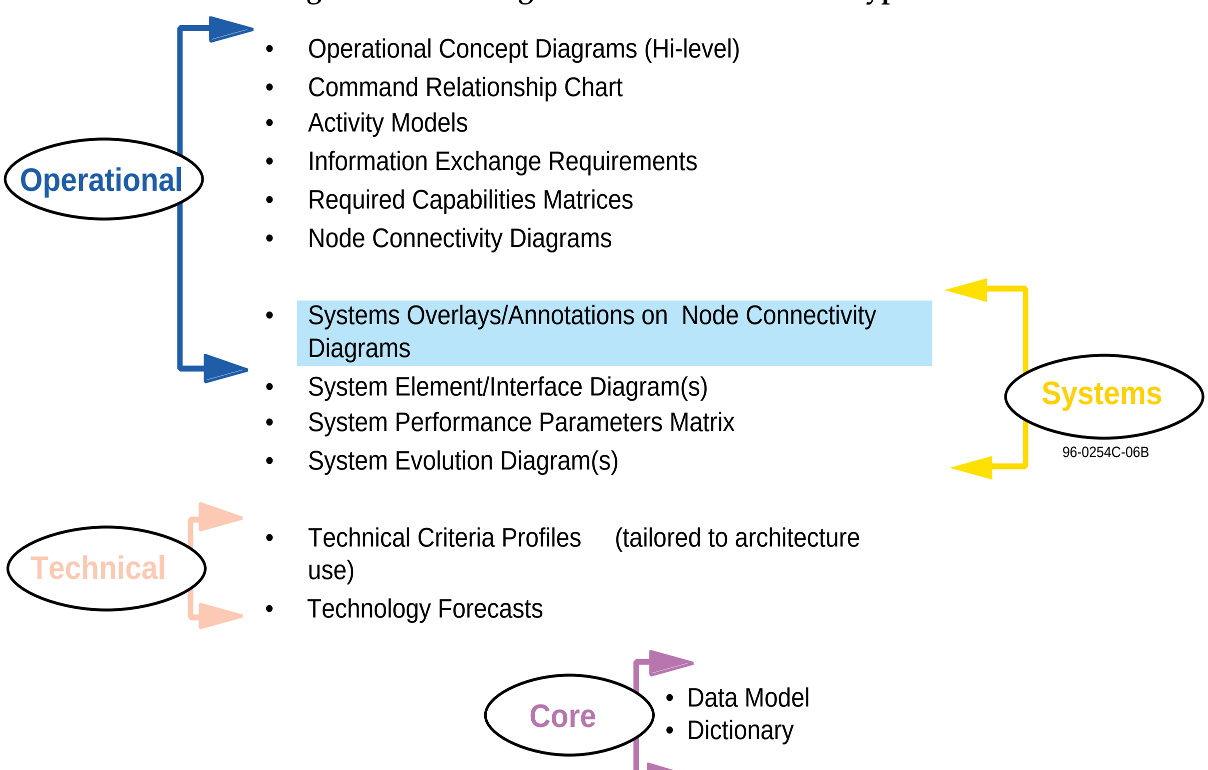


Figure 3-11: Initial Set of Architecture Products

This list is not meant to imply that every architecture of any particular type must include the full list of its associated products. Instead, individual products from the list should be developed as they support the objectives of a specific architecture effort. In addition, the level of detail to which any particular product needs to be developed varies depending upon the specific objectives that the architecture is designed to meet.

3.4 USING THE FRAMEWORK

3.4.1 Architecture Development Process

The following series of steps should be followed in developing an architecture:

1. Determine the intended use of the architecture (e.g., document capabilities, assess issues)
2. Determine architecture's scope and context to include any assumptions or constraints to be considered
3. In accordance with the above, determine what specific architecture characteristics must be captured or displayed
4. Based on the characteristics to be displayed, determine which types of architecture products should be built. To meet any specific objective, products representing one, two, or all three types of architectures may need to be developed. Also, for each type of architecture one, two, or all of the constituent products may be required. In some cases only selected products are needed.
5. Build the products and use the architecture.

Additional information on the architecture development process is provided in Annex B.

3.4.2 Use of Architecture Products

Figure 3-12 (on the next page) shows an example of how different architecture products can be used to answer sample questions often posed of architectures. As can be seen in this figure, it may take several products, often representing different types of architectures to answer a particular question. Also any given product can often be used to answer more than one kind of question.

Sample Questions	Typical Applicable Architecture Products		
	Operational	Systems	Technical
Who needs the information?	Operational Concept Diagrams Command Relationship Charts		
Who produces the information?	Operational Concept Diagrams Command Relationship Charts	Supporting Systems Diagrams/Descriptions	
Do the means exist to convey the information from the producers to the users?	Operational Concept Diagrams Node Connectivity Diagrams	System Data Flow Diagrams	Information Standards Descriptions
Do systems have the set of enabling functions necessary to conduct the required information transactions?	Activity Diagrams/ Data Models	Systems Overlays/ Annotations on Node Connectivity Diagrams Activity Allocation to System Component Descriptions	
Are the enabling functions implemented in accordance with approved DoD-wide standards, conventions, product-based solutions, etc.?		Supporting Systems Diagrams/Descriptions	Information Standards Descriptions

Figure 3-12: Sample Use of Architecture Products

SECTION 4
PRODUCT DESCRIPTIONS

4.1 GENERAL

This section describes each of the architecture product types referenced in Section 3. For most of the types, a generic “template” is shown that illustrates the basic format of the product, describes the characteristics to be captured in the product, and lists some of the uses of the product. Additional examples are provided in **Annex E**. For any architecture effort, the specific products to be produced and the level of detail to which they are developed depends on the purpose of the architecture.

4.2 Operational Architecture Products

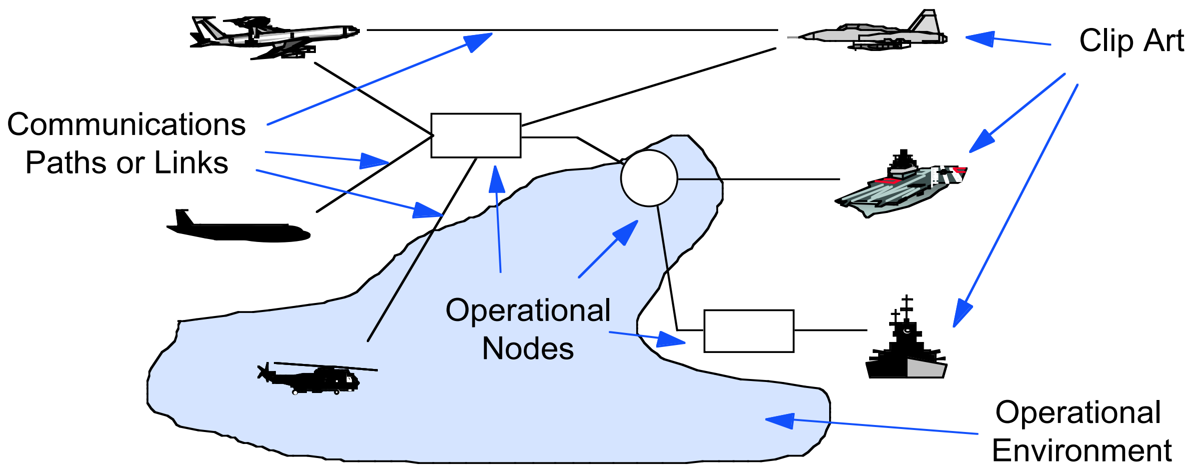
Standard products associated with operational architectures focus on the warfighting context for C4ISR support, the missions and tasks to be supported, the operational elements involved in accomplishing the tasks, and the information exchanges needed to meet operational needs. Products include:

- High-Level Operational Concept Diagrams
- Command Relationship Charts
- Activity Models
- Information Exchange Requirements
- Required Capabilities Matrices
- Node Connectivity Diagrams

The central theme common to these products, as identified in the Integrated Architecture Panel architecture definitions, is the *information flow* that links operational elements and the activities required to accomplish operational missions. To facilitate applicability across Services, commands, and DoD agencies, the content of operational architecture products is keyed to the warfighting and warfighter support missions and tasks described in the JCS’s Universal Joint Task List (UJTL) and task lists derived from the UJTL such as Joint Mission Essential Task Lists (JMETLs) or Service-specific task lists (e.g., Navy Mission Essential Task List [NMETL]).

4.2.1 Operational Concept Diagram

The Operational Concept Diagram, as depicted in **Figure 4-1** (on the next page), is used to depict the “big picture” view of the operational warfighting context. It is aimed at senior-level decisionmakers and uses a graphical picture to represent a high-level view of the operational



Features:

- High-level description of operational concept
- Graphic portrayal oriented to senior-level decisionmakers

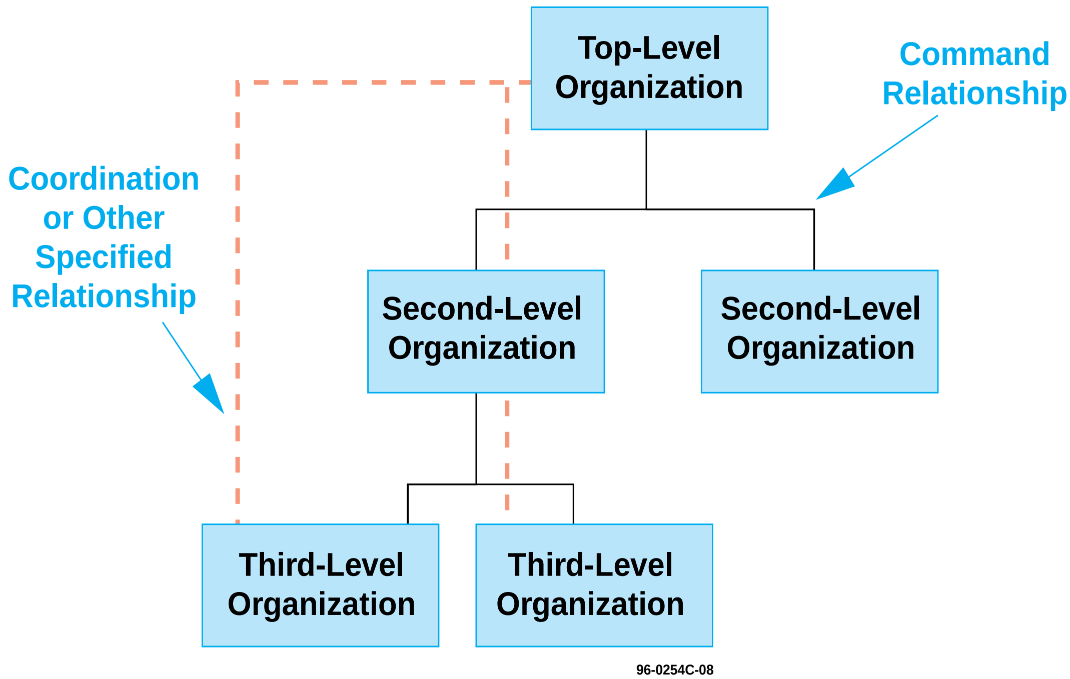
Figure 4-1: Operational Concept Diagram

environment in terms of operational elements or echelons involved, geographic region, nodal connectivity, types of forces employed, etc.

The figure shows generic icons that can be tailored as needed and used to represent various classes of players in the architecture, e.g., an aircraft icon can represent a particular type of aircraft, or a particular air organization, or the air assets of a joint task force. The icons can also be used to represent missions or functions, e.g., the aircraft icon could represent Air Operations and the ship icon could represent Maritime Operations. The lines connecting the icons can be used to show simple connectivity, or can be annotated to show what information is exchanged. How the template is tailored depends on the scope and intent of the architecture, but in general, an Operational Concept Diagram will show such things as the missions, high-level functions, organizations, and geographical distribution of assets.

4.2.2 Command Relationships Chart

The Command Relationships Chart, as depicted by the template in **Figure 4-2** (on the next page), is used to show the operational elements involved in a particular military operation and the relationships among them. It depicts lines of command and coordination among operational elements and may depict operational elements in either generic terms or by particular organizational element, depending on the particular need. The level of detail to be shown on this chart is commensurate with the intended use of the architecture. This type of chart should be drawn only to the level that depicts the applicable operational elements and lines of command.



Features:
- Identifies command, control, and coordination relationships among operational elements organizations
- Level of detail commensurate with intended architecture use

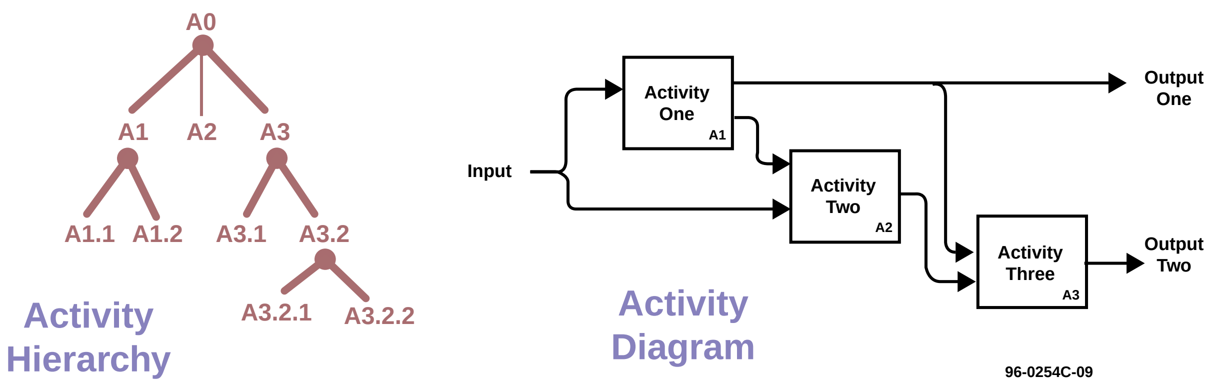
Figure 4-2: Command Relationships Chart

Command relationships may be important to show in an operational component of an architecture because they illustrate "how business is done." For example, command and control relationships may differ under different circumstances, such as for various phases of warfare; these command structures may mean that activities are performed differently or by different units. Different coordination relationships may mean that connectivity requirements are changed.

4.2.3 Activity Models

As shown in **Figure 4-3** (on the next page), activity models describe the applicable activities associated with specific warfighting tasks that must be accomplished to support a particular mission, the relationship among activities, the data or information exchanged between activities, and the data or information exchanged with other activities that are outside the scope of the model. The models are hierarchical in nature, i.e., they begin with a single box that represents the overall activity and proceed successively to decompose the activity to the level required by the purpose of the architecture.

4.2.3.1 Associated Diagrams. Activity models include two different types of diagrams: the activity hierarchy or "tree" and the activity diagram.



Features:

- Keyed to warfighter (e.g., UJTL) activities
- Activities assigned to nodes (organizations, facilities, workstations, etc.), depending on level of architecture

Figure 4-3: Operational Activity Diagram

The activity hierarchy or "tree" shows which tasks are decomposed from others. The hierarchy of tasks presented in the UJTL represents this kind of relationship. For any given operational architecture, the activity hierarchy should start using one of the branches of the UJTL as the top of the tree. The top-level task should be broken down into lower level tasks until the tasks at the lowest level can be clearly identified with a particular operational element or node responsible for accomplishing the task. The activity tree should be refined only to the level necessary to meet the needs of the particular operational architecture being developed. Depending on the specific objectives to be served by the operational architecture, it may be appropriate for some branches of the tree to be refined to very low levels whereas others are defined only at high levels. In many cases, higher level tasks are presented only to show the context within which the lower level tasks are performed.

The activity diagram shows the relationship among the tasks at any given level of the activity hierarchy. The objective of the activity diagram is to show dependencies among activities, principally with respect to the information flows among them.

Activity models can capture valuable information about an architecture and can promote the necessary common understanding of the domain under examination. However, care must be taken to make sure that the modeling process is performed efficiently and usefully. An approach that CISA has advocated and has used successfully is the template model approach. Using this approach, an activity model template is constructed and used as a guideline for building multiple models that cover the same set of activities but from different viewpoints. The model specifies the activities, generic input/output/control/mechanism categories, and specific characteristics to be captured in the model. The different viewpoints can be those of multiple organizations that perform similar activities; in that case, the template approach allows those organizations' processes to be easily compared. With or without the incorporation of template models, it is often useful to construct a high-level, generic model of the subject in question and then to build a number of related models of various aspects of that subject area. The objective in any of these techniques is to focus the modeling effort so that a number of small, quickly developed models can be used instead of a large, many-layered model.

Activity models generally include a chart of the hierarchy of activities covered in the model, a facing-page text for each diagram to provide any required detail, and a data dictionary that defines all activities and terms used in the diagrams.

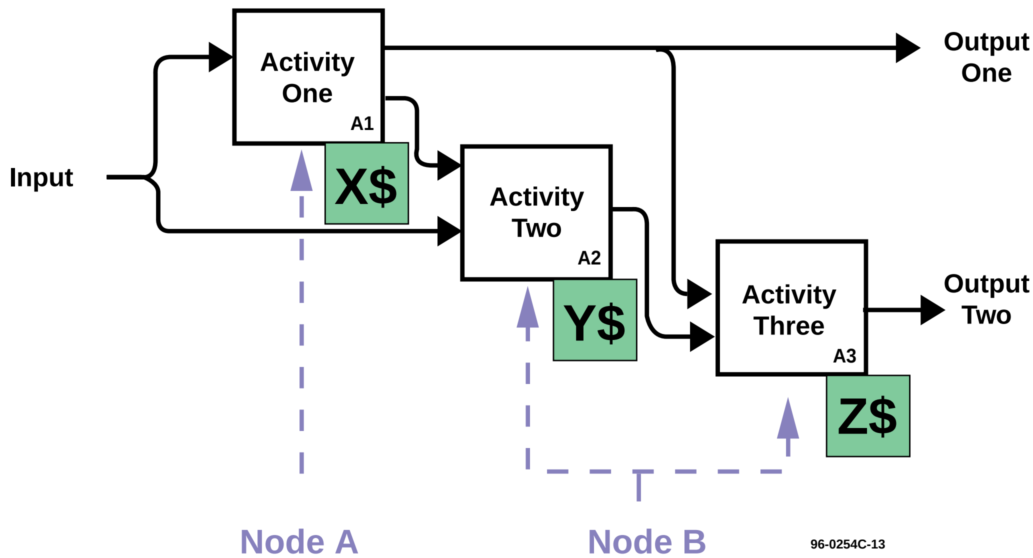
4.2.3.2 IDEF0 Modeling Language. IDEF0 * is a frequently used construct for depicting activity relationships in diagram form. DoDD 8020.1-M, Functional Process Improvement (FPI), was issued as interim guidance in January 1993 and specified the use of IDEF in FPI analysis. IDEF0 can be very useful for providing indepth understanding of functional activities and as such may be considered background analysis in support of operational architecture development. IDEF0 provides a sound methodology for activity modeling, but it is not the only construct that may be used. Other activity modeling conventions, such as DeMarco diagrams, may also be used. The main point to keep in mind is that the purpose of the activity diagram is to depict the relationship among the activities particularly with regard to information inputs and outputs.

The precise structure of IDEF0 activity modeling, especially in view of its widespread acceptance throughout DoD, imposes a degree of standardization in the diagrams that facilitates common understanding of the presentation of the operational architecture information. A key feature of the IDEF methodology are the accompanying integrated dictionaries that provide precise definitions of activities, their inputs and outputs, the mechanisms that support activities, and the controls that guide how activities should be accomplished. The IDEF methodology also provides a logical progression from IDEF0 activity modeling to IDEF1X data modeling, which is necessary to understand the relationships among the elements of warfighter information that are the subject of the C4ISR architectures. Data models of warfighter information provide the logical basis for designing and developing information processing systems in support of operational needs. Although not always required, when they are developed, the IDEF1X data models of warfighter information are considered a core architecture product in the Framework construct. Additional information on IDEF modeling may be found in DoD 8020.1-M, Functional Process Improvement, January 1993.

4.2.3.3 Overlays to Activity Models. One way to get the most out of a relatively small activity modeling effort is to overlay additional information onto the basic diagrams in order to gain greater insight without additional decomposition. Nodes that perform an activity can be indicated on the appropriate activity box. (This kind of annotation is a standard part of the IDEF0 methodology, and is used in the preceding example. This kind of annotation could also be

* IDEF0 is the activity modeling technique associated with the Integrated Definition (IDEF) language.

added when other methodologies are used.) Costs of performing the activity can be indicated, and specific attributes of exchanged information can be added to the arrow labels. If such annotations and overlays are designed carefully, the purposes of the architecture can be furthered with relatively little extra effort. **Figure 4-4** is a template showing some sample overlays.



Features:
- Activity Model serves as template
- Variety of data may be overlayed on template (e.g., nodes, costs)

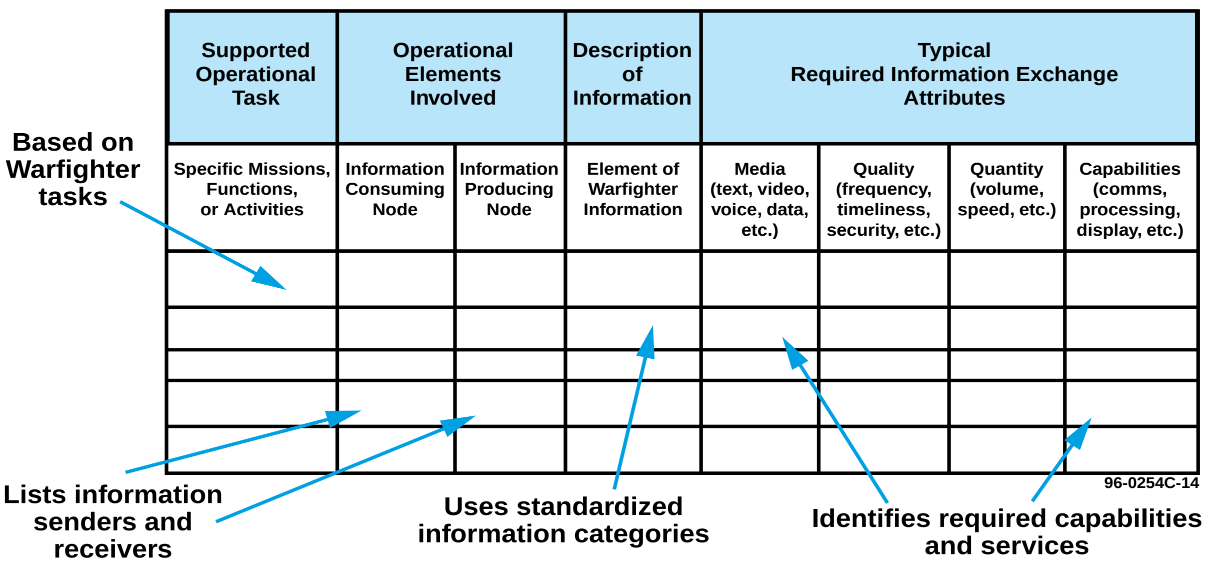
Figure 4-4: Activity Model Overlay

The dashed arrows indicate which nodes perform which activities; this information can be used to uncover unnecessary functional redundancy. What constitutes a “node” will depend on the level of the architecture being built and its purpose: in some cases a node will be an organization, in others a node will be a facility or even an individual workstation. The dollar signs indicate that the costs of performing an activity could be appended as well; this activity-based cost information can be used to make decisions about streamlining, combining, or omitting activities.

4.2.4 Information Exchange Requirements

As defined in Section 3, Information Exchange Requirements (IERs) express the relationship across the three basic entities (tasks, operational elements, and information flow) in an operational architecture. Using the defined activities as a basis, IERs identify the elements of warfighter information used in support of a particular activity and between any two activities. IERs identify *who* exchanges *what* information with *whom*, *why* the information is necessary and in *what* manner. The information media (i.e., data, voice, and video), quality (i.e., frequency, timeliness, and security), and quantity (i.e., volume and speed) are attributes of the information exchange that may be included in the IER. Particular capabilities such as automated data processing capabilities, secure communications, facsimile, database query, and large-screen display, may also be captured for each exchange. Required capabilities may be extended to capture other needs such as personnel skills or facilities. The specific attributes which are used to describe the information exchange are driven by the purpose for which the architecture is being developed.

Figure 4-5 illustrates a matrix approach for representing IERs. The starting point for defining IERs is the activity for which an information exchange is being defined. The activity should be traceable to a specific UJTL task. The matrix is designed to portray each information exchange on its own separate line. Consequently, multiple rows of the matrix may be required to describe all of the information exchanges needed to accomplish any particular activity, and the overall size of the Information Exchange Matrix may become quite voluminous. Fortunately, due to its highly structured format, the Information Exchange Matrix lends itself readily to linkage to relational databases from which the matrix can be generated automatically. In practicality, hard copy versions of this architecture product should be limited to high-level summaries of categories of information exchange or highlighted subsets of particular interest.



Features:

- Focuses on information content, flows, and supporting services required to accomplish operational tasks
- Level of detail shown depends on purpose of architecture

Figure 4-5: Information Exchange Matrix

As shown in the figure, moving across the matrix, there are separate columns to describe the information that is being exchanged, identify the operational elements or nodes that use and originate the information, and define the relevant required attributes of the information exchange such as the media, quality, quantity, and capabilities associated with the information exchange.

The scope and purpose of a particular architecture will determine the degree of specificity with which the information and the consuming/producing nodes are described and the specific attributes that are used to describe the IER. For example, for high-level planning architectures, it may only be necessary to identify information exchanges with respect to major categories of information and primary operational nodes involved in the exchange with only a few high-level attributes such as media and security described. To support development of systems architectures, however, it will likely be necessary to provide greater specificity as to the information content, consuming/producing nodes, and media as well as fairly explicit qualitative and quantitative descriptions of the information exchanges and descriptions of particular types of processing and communications services that are required.

The elements of information should be described in common terms that can be readily understood throughout the C4ISR community. An attempt to develop a standardized list of categories of information is underway as part of refinement of the framework.

4.2.5 Node Connectivity Diagram

Once nodes have been paired with activities , the connectivity required to perform those activities can be illustrated. The Node Connectivity Diagram depicted in **Figure 4-6** presents a visual portrayal of the nodes, activities, and the connectivity required to perform those activi-

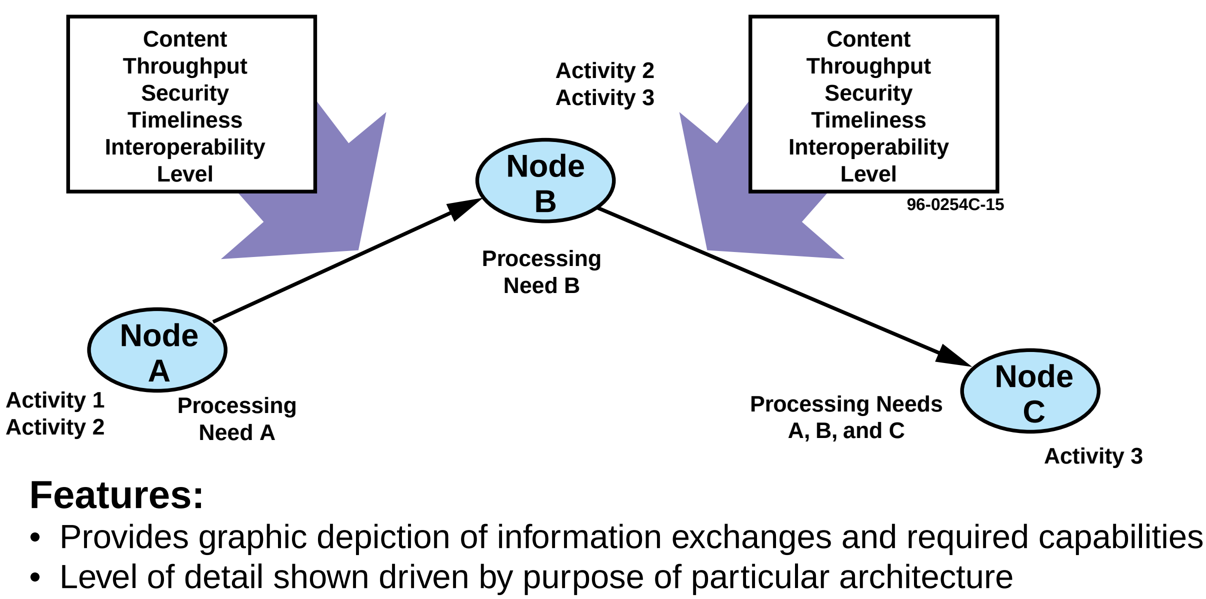


Figure 4-6: Node Connectivity Diagram

ties. The Basic Node Connectivity Model in effect “turns the activity model inside out,” focusing on the physical nodes rather than on the abstract activities. The main features of this kind of diagram are the nodes, the needlines between them, and the characteristics of the information exchanged. Each IER is represented by an arrow, which is annotated to describe the characteristics of the data or information, i.e., its substantive content, format (voice, imagery, text and message format, etc.), throughput requirements, security or classification level, timeliness requirement, and the level of interoperability required for the exchange. The activities associated with a given IER can be noted alongside the node or on the arrow, in order to allow functional solutions, rather than systems solutions, to be discovered.

The Node Connectivity Diagram can be used to depict required communications capacities of different types of information (e.g., data, voice, video) between nodes, or particular services (e.g., database access, large screen display) required at different nodes.

The information illustrated in node connectivity models can be used to make decisions about what systems are needed to satisfy the business needs of an organization or functional area. However, it is the *business needs* that are illustrated, not the systems solutions; therefore, this kind of diagram is included in the operational component rather than in the systems component. No systems have been named yet, except in the very broadest sense, in which a node could be considered a “system.”

Except in the case of rather simple situations, any attempt to present the entirety of the required capabilities in Node Connectivity Diagram form will likely detract from its readability and defeat its purpose of facilitating comprehension by the operator or functional advocate for the operational architecture.

4.2.6 Required Capabilities Matrix

The Required Capabilities Matrix, presented in **Figure 4-7** (on the next page), summarizes the capabilities (quantitative and qualitative requirements and services) that are required at a particular node or between any two nodes. As indicated in the figure, the capabilities required at any single node are cataloged in the matrix cells that lie along the diagonal, while required capabilities between cells are cataloged in the off-diagonal cells. The contents of the Required Capabilities matrix may be obtained by summarizing the contents of the information exchange matrices by node.

Normally, the capabilities required between two nodes do not depend on the direction of data exchange. However, occasionally the need exists for documenting “one-way” requirements, such as would be the case for broadcast communications. As can be seen in the template, since there are two cells available for summarizing the required capabilities between any two nodes, the matrix offers the flexibility of cataloging two-way requirements as well as separate one-way requirements in each direction. In particular, since information generally flows into, as well as out of, nodes, the boxes in the lower diagonal can be used to describe information inputs while the upper diagonal can be used to describe outputs.

- Describes required capabilities (e.g., communications, interoperability) between any two nodes or at single nodes (e.g., processing)
- Separate charts used to focus on particular categories of requirements

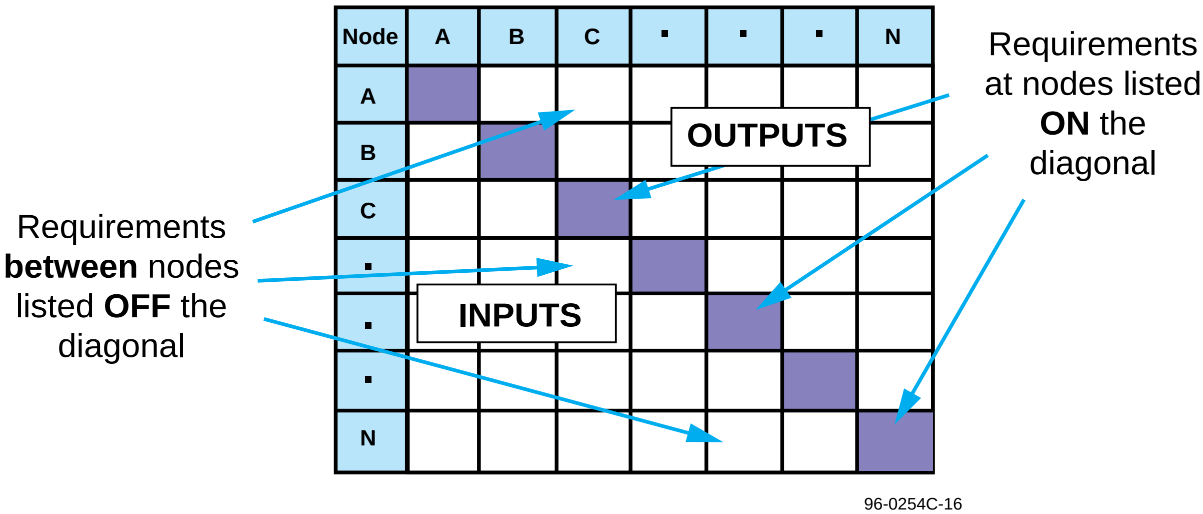


Figure 4-7: Required Capabilities Matrix

Because the Required Capabilities Matrix simply represents a summary by node and by node pair of the information contained in the Information Exchange Matrix, use of a relational database to maintain the contents of the Information Exchange Matrix database would permit easy rapid generation of the Required Capabilities Matrix. The "input-process-output" format of the matrix is consistent with activity modeling and provides a convenient way to summarize some of the information in activity models.

4.3 SYSTEMS ARCHITECTURE PRODUCTS

The systems component products are largely derived from the operational component products and are therefore dependent on the operational component products for their focus and level of detail. Warrior ownership of the operational component products provides the driving operational vision, while the C4ISR architect uses the systems architecture products to provide the necessary systems and connectivity support to that vision.

4.3.1 Systems Overlays

Systems Overlays assign specific hardware/software systems to the nodes described in the Basic Node Connectivity Model. These systems assignments are shown as overlays to the Basic Node Connectivity Model. Depending on the focus of the architecture, these systems can include automated information systems, communications systems, or others. In addition, communications nodes are depicted, to trace the path of an IER from its source to its ultimate destination, as are the required services. The systems information captured in these overlays can be used to compare systems used at various nodes in order to identify opportunities to improve performance or eliminate redundancies by making system changes.

A template for a Systems Overlay is shown in **Figure 4-8**; this is a modification of the template for the Basic Node Connectivity Model.

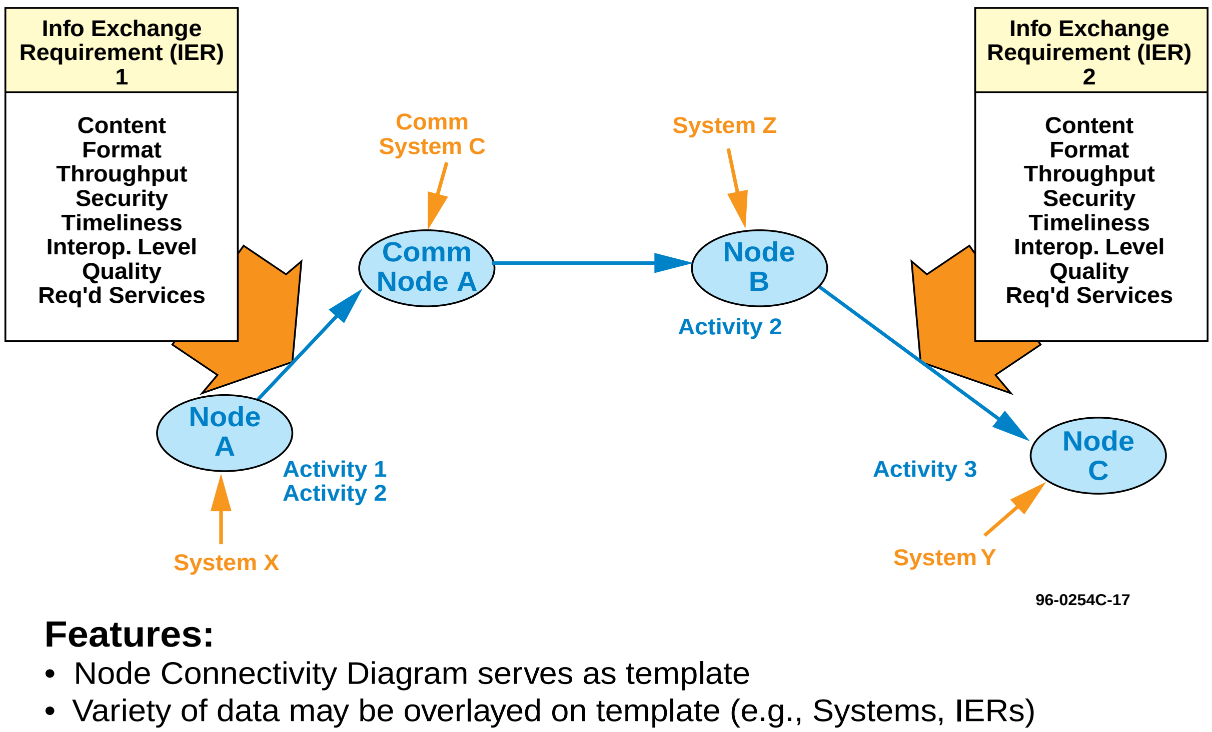


Figure 4-8: Systems Overlay

4.3.2 System Element/Interface Diagram

A System Element/Interface Diagram decomposes the nodes from the Node Connectivity Model with System Overlays to reveal the relationships among the systems resident at the nodes. The systems, their configurations, and their linkages are shown on the diagram. A table is constructed that shows the relevant data/information exchanges between systems within the node and the data/information exchanges between systems at the node and systems at other nodes (external exchanges). For each system, the inputs, processing, and outputs of the system's hardware elements and applications are listed. The detailing of the external exchanges allows an IER to be analyzed to determine which systems at which nodes contribute which data elements of an IER. This kind of information can be used to analyze and improve the configuration of systems and local area networks (LANs); to determine more efficient distribution of software applications; and, in conjunction with the System Performance Parameters Matrix and portions of the technical architecture component, to examine interoperability problems. A notional example of a System Element/Interface Diagram is shown in **Figure 4-9**.

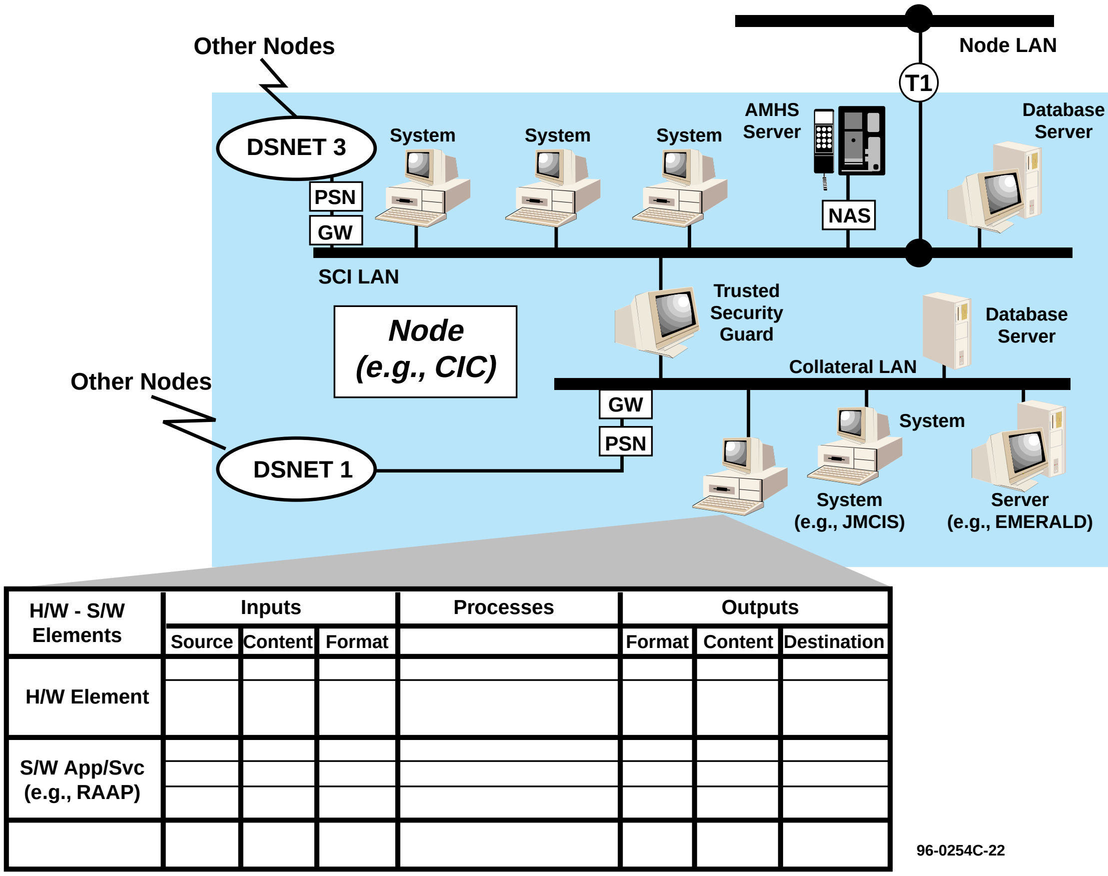


Figure 4-9: System Element/Interface Diagram Example

4.3.3 System Performance Parameters

The System Performance Parameters Matrix builds on the System Element/Interface Diagram to describe the current performance characteristics of each system, and the expected or required performance characteristics at specified times in the future. Characteristics are listed separately for the hardware elements and the software elements. The future performance expectations are geared to the Technology Forecast portion of the technical architecture component. **Figure 4-10** (on the next page) shows an example of a System Performance Parameters Matrix, listing representative performance characteristics.

4.3.4 System Evolution Diagram

The System Evolution Diagram describes plans for evolving a suite of systems into a more streamlined, efficient (smaller and cheaper) set. It builds on the previous diagrams and analy-

System Name:	Performance Thresholds/Measures		
	Time$_0$ (Baseline)	Time$_1$	Time$_N$ (Objective)
Hardware Element 1			
Mean Time Between H/W Failures/Faults			
Maintainability			
Availability			
System Initialization Time			
Database Transfer Time			
Program Restart Time			
Hardware Element N			
Software Element 1			
Data Capacity (e.g., throughput or # of input types)			
Automatic Processing Responses (by input type, # processed/unit time)			
Operator Interaction Response Times (by type)			
Effectiveness			
Availability			
Mean Time Between S/W Faults			
Organic Training			
Software Element N			

Figure 4-10: System Performance Parameters Matrix Example

ses in that information requirements, performance parameters, and technology forecasts must be accommodated. An example System Evolution Diagram is shown in **Figure 4-11**.

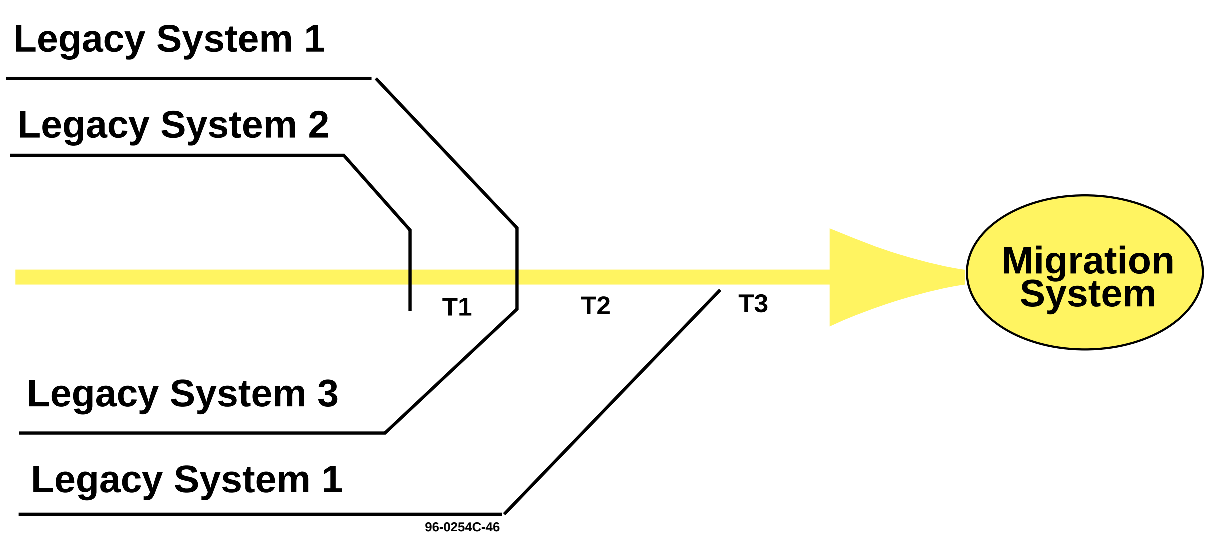


Figure 4-11: System Evolution Diagram Example

4.4 TECHNICAL ARCHITECTURE PRODUCTS

As defined earlier, the technical architecture provides the technical guidance that governs system implementation and operation. There are a number of technical reference models (TRMs) in existence that can serve as sources for technical guidelines, such as the DoD Technical Architecture Framework for Information Management (TAFIM), the Joint Technical Architecture and the Army Technical Architecture. In addition, some descriptions of information standards are available, such as the Defense Data Repository system (DDRS) and the C2 Core Data Model.

4.4.1 Tailored Technical Criteria Profile

In many cases, especially in building architectures with less than a Service-wide scope, “building” a technical architecture will really consist of identifying the applicable portions of existing technical guidance documentation, tailoring them as needed, and filling in any gaps. The Tailored Technical Criteria Profile is a product that helps to capture the technical guidelines applicable to a given architecture. The profile is time-phased to facilitate a structured, disciplined process of system development and evolution. Time-phasing also promotes the consideration of emerging technologies and the likelihood of current technologies and standards becoming obsolete.

A Tailored Technical Criteria Profile constructed for a given architecture will be structured as appropriate in accordance with the purpose for which the architecture is being built. For example, an architecture may be built to examine issues of information system interoperability. In that case, the Tailored Technical Criteria Profile could be organized around the applicable levels of interoperability. The Tailored Technical Criteria Profile would show the enabling functions and criteria for each of the required levels of interoperability. An example of such a Tailored Technical Criteria Profile is shown in **Figure 4-12**. (Levels of Interoperability are discussed in Section 4.6. Levels shown here are for illustration only.)

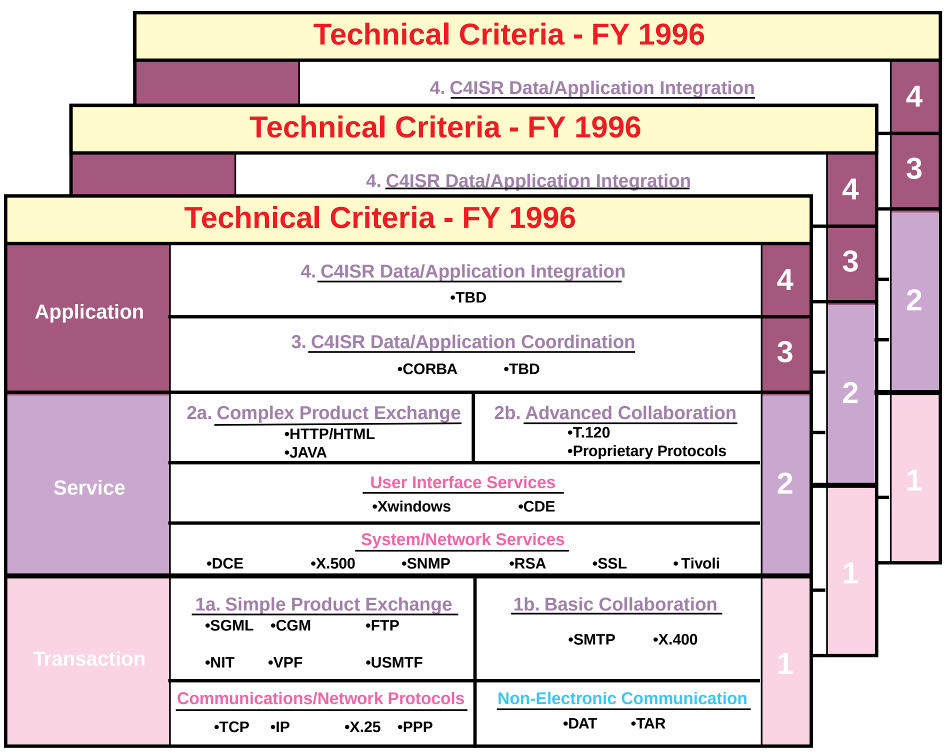


Figure 4-12: Tailored Technical Criteria Profile Example 96-0254C-23

4.4.2 Technology Forecast

A Technology Forecast is a detailed description of emerging technologies and specific hardware and software products. It contains predictions about the availability of emerging capabilities and industry trends in the 6-month, 12-month, and 18-month time frames, and confidence factors for the predictions. The forecast includes potential technology impacts on current architectures, and thus influences the development of transition and objective architectures. The forecast should be tailored to focus on technology areas that are related to the purpose for which a given architecture is being built, and should identify issues that will affect the architecture. **Figure 4-13** depicts a sample Technology Forecast focused on the area of data production and management.

Technology Domain: Data Production and Management

Forecast

Technology Areas & Capabilities	*Short Term 0-6 Months*	*Mid Term 6-18 Months*	*Long Term 18+ Months*
Forecast of Industry Developments			
Dist. Heterogeneous DBs	• Middleware and/or proprietary interfaces • CGI-BIN connections to Web	• KQML • Development of APIs for Web access • Updates Dynamic using Java	
Security	• Limited RSA & Significant OS Level	• COTS RSA	• Fortezza & RSA
Hyperlink Management	• Limited Tools	• Wider availability of better tools	• Intelligence Agents
Document Creation Tools	• SGML, HTML — WWW, VRML	• SGML, HTML, VRML	• SGML, Java, VRML
Formats	• GIF, JPEG, PDF, Java (Netscape)		• Universal w/NITF
Data Management	• Middleware Dependent		• Transparent to User
Throttling Capability		• Firewalls	
Data Replication		• Network Mirroring	

Issues

- *Hyperlink Management:* Strong policies needed
- *Profiling & Push:* Expanded push instead of Netscape's Server Push
- *Throttling:* Who gets to what servers when, maybe a policy issues first
- *Security:* Transition Issue (Fortessa) — Netscape has said they will add Fortessa; what does DoD need to do if others do not follow suit?
- *Interfaces with NIDR search tools*

Figure 4-13: Technology Forecast Example

4.5 CORE INFORMATION PRODUCTS

The data model and data dictionary are considered core products because they relate to all three types of architectures. There are two types of data sets associated with architectures: architecture information and warfighter information.

Architecture information are the things that are described in architectures. These include all those terms associated with the three types of architectures such as missions, tasks, operational elements, nodes, systems, services, and standards. **Annex A** contains an initial version of a entity-relationship (E-R) data model for architecture information. The Annex A example focuses primarily on information associated with operational architectures but the approach is applicable to all three architecture types.

Warfighter information is that data required by the warfighter to accomplish tasks. It is that data to which the IER refers in the operational architecture, the data which are being passed between systems in the systems architecture, and the data for which data element standards are defined in the technical architecture. A strawman taxonomy of warfighter data is provided in **Annex D**.

4.5.1 Data Model

The data model describes the data and the relationship between data elements. A common approach is to describe the data in terms of entities and relationships. Entities are objects that exist and are distinguishable from other objects. A relationship is an association among entities.

Data models may be developed for architecture information and for warfighter information. A separate model would be developed for each of these two types of data.

Figure 4-14 is a template for a data model.

4.5.2 Data Dictionary

The data dictionary is a repository of information about data such as definition, relationships to other data, origin, usage, and format. It provides a common reference point for the definition of information, documents information structures, and identifies standard data elements. Data dictionaries may be developed for architecture information and for warfighter information.

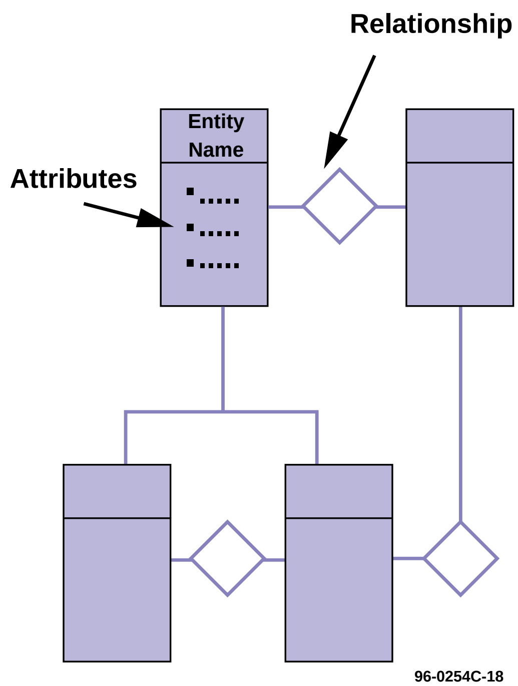


Figure 4-14: Data Model

In Section 3, the data dictionary is included as an essential part of a technical architecture because of the critical role that standard data elements play as part of system standards. Standard data elements are the fundamental interoperability enabling feature of all systems; they are the basis for common interpretation, processing, the display of information, and communications efficiency. The data dictionary is considered a core product because of its applicability to operational and systems architectures as well as technical architectures. Data dictionaries of warfighter data are important tools in understanding the information flows within the operational and systems architectures. Data dictionaries of architecture data apply equally to all architecture types.

Data dictionaries are necessary to support the understanding and sharing of information among different communities, and greatly simplify the development of architecture features. The data characteristics contained in the dictionary are used to design, monitor, document, protect, control, and understand the data in an automated system. They support information reuse, resulting in reduced data duplication, reduced costs, and increased data consistency. Because architectural requirements constantly change, and a change in one part usually affects other parts of the architecture, a repository also enhances configuration management procedures.

4.6 LEVELS OF INTEROPERABILITY CONCEPT

The Integrated Architectures Panel has endorsed the concept of standard descriptions for levels of information system interoperability. The concept is currently a work in-progress. When finalized, the incorporation of the Levels of Interoperability into the architecture development process will better clarify interoperability capabilities and requirements across systems. This concept can be associated with several of the architecture products discussed in this section and can serve as an integrating mechanism among architecture components. The level of node-to-node interoperability required (per IER) is defined during the process of developing the Basic Node Connectivity Model of the operational architecture component. In building the Systems Overlays to the Basic Node Connectivity Model and the System Element/Interface Diagrams, the node-to-node interoperability requirements are translated into required levels of interoperability between systems. The current version of the levels and their meanings are illustrated below in **Figure 4-15** (on the next page).

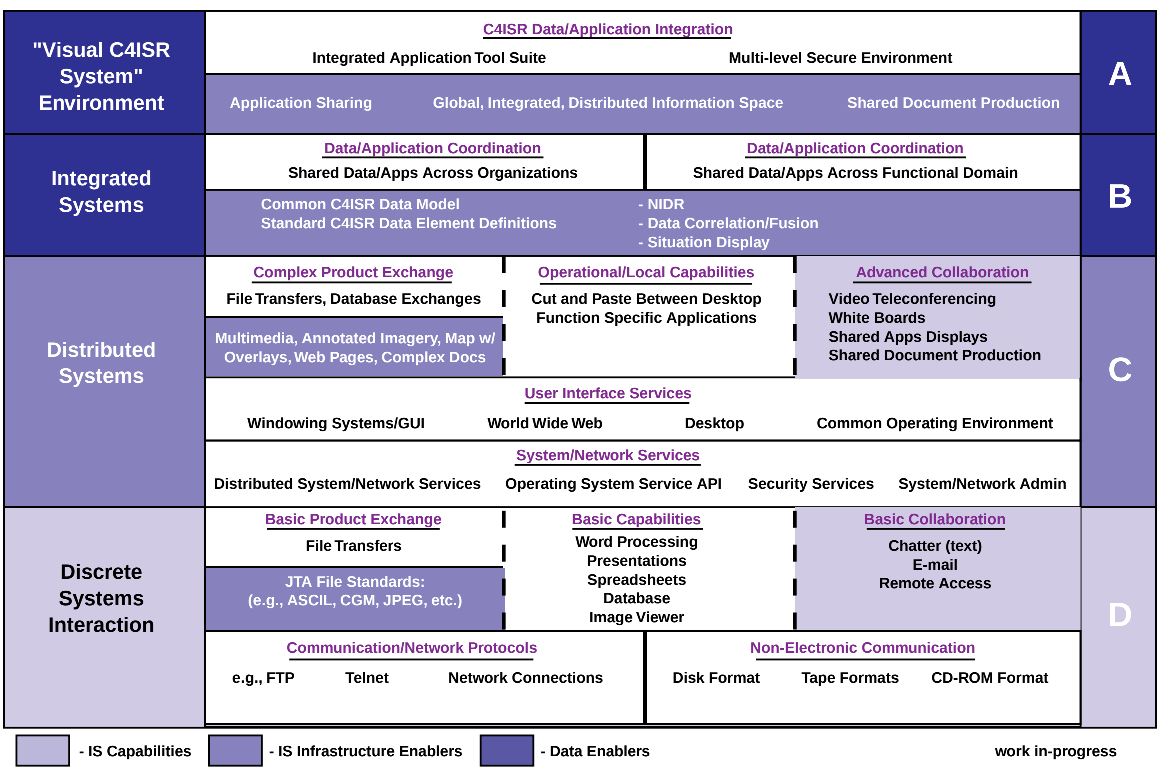


Figure 4-15: Levels of IS Interoperability Construct—May 1996 96-0254C-24

This Page Intentionally Left Blank

ANNEX A

ARCHITECTURE INFORMATION ENTITY-RELATIONSHIP DATA MODEL

An entity-relationship (E-R) data model shows how the sets of information contained in architectures are related to each other. Like the wooden forms used in the building trade to hold concrete until it sets, the architecture information data model is not part of any architecture. As such, it should not be confused with the data models that are part of C4ISR architectures, such as E-R diagrams that show how various elements of warfighter information are processed to support a particular task or activity.

Figure A-1 addresses the information contained in the Operational Architecture and shows relationship to systems. This diagram will be expanded in the next version of the Framework report to more fully address systems and technical architecture data sets.

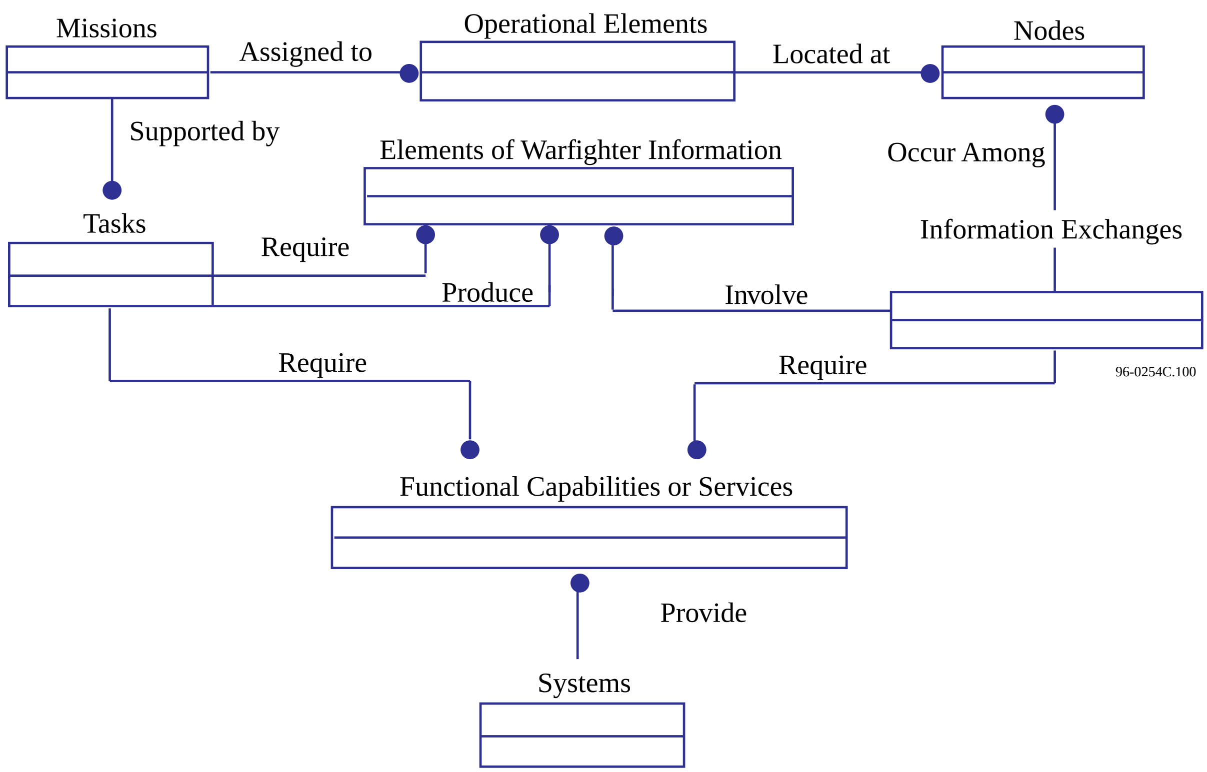


Figure A-1: Data Model of Architecture Information

This Page Intentionally Left Blank

ANNEX B

DEVELOPING PRODUCT-BASED ARCHITECTURES

B-1 Guiding Principles Revisited

The set of Guiding Principles was initially introduced in Section 3. The following provides some additional considerations on those principles. The set of guiding principles for building architectures has purposely been kept small; therefore, each is critical to the objectives of the guidance. The principles are listed below.

1. *Architectures should be built with a purpose in mind*. Having a specific and commonly understood purpose before starting to build an architecture greatly increases the efficiency of the effort and the utility of the architecture. The purpose determines the appropriate scope, the characteristics that need to be captured, and the time phases that need to be considered. This principle applies equally to an architecture as a whole and to any portion of an architecture. It can also be said to apply to groups of architectures. If groups of architectures built by various organizations are to be compared, it is important that they all be built from the start with the purpose of comparison in mind.

2. *Architectures should facilitate, not impede, communication among humans*. Architectures must be structured in a way that allows humans to understand them quickly and that guides the human thinking process in discovering, analyzing, and resolving issues. This means that extraneous information must be excluded (see principle number one) and common terms and definitions must be used. Often, graphical formats are best for rapid human understanding, but the appropriate format for a given purpose must be used, whatever that format may be.

3. *Architectures across DoD should be relatable, comparable, and integratable*. Like principle number two, this one necessitates the use of common terms and definitions. This principle also necessitates that a common set of activities be used as the basis for architectures. A likely candidate for this common set of warfighter and warfighter-support activities is the Universal Joint Task List (UJTL). However, the UJTL is only a list as opposed to a model (for example, a model that shows standard inputs and outputs to each of the activities and that assures input/output consistency across all levels of all activities) and is only detailed to a finite level of decomposition. Therefore, some architectures may need to use different activities or decompositions of activities; however, deviations from the UJTL should be mapped to any corresponding activities within the UJTL.

   This principle also dictates that similar-type products developed for different architectures display similar types of information about their respective domains, in similar formats. (This is discussed further in Section 4.)

4. *Architectures should be modular and reusable.* Architectures should consist of separate but related pieces that can be recombined with a minimum amount of tailoring, so that they can be used for multiple purposes.

A fifth principle could be added to the initial four:

5. *Architecture efforts should be designed to obtain the most useful results in the least amount of time.* It should not always be necessary to spend large amounts of time, money, and resources in order to obtain useful results from an architecture development/analysis effort.

The recommended characteristics (information) to be captured, the set of products to be built to capture those characteristics, and the procedure for using the Framework have all been designed to ensure that the above principles are followed. These aspects of the Framework and some guidance for documenting architectures are discussed in following paragraphs.

B-2 The Role of Products

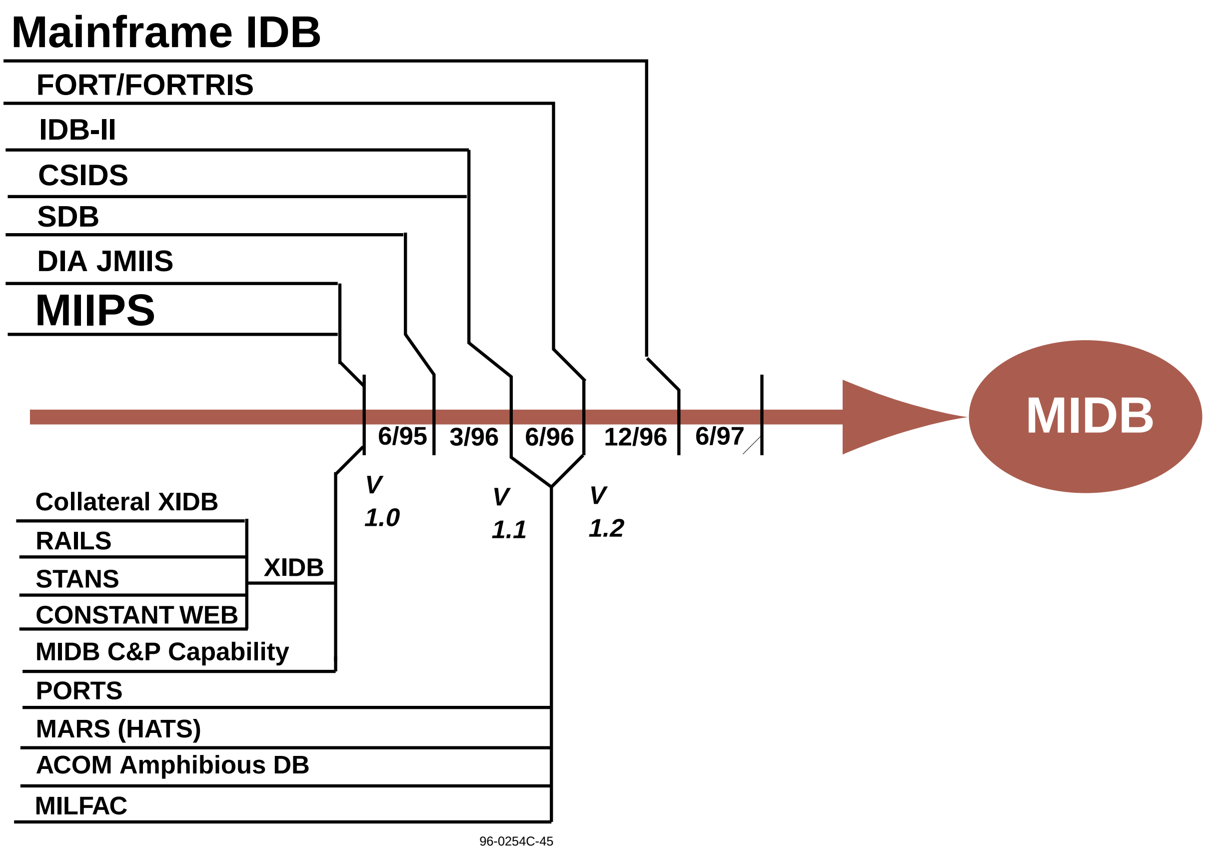
When completed for a given architecture, the set of products constitutes the architecture. These architecture products are distinguished from preexisting information sources that may be used in building architectures, such as existing models, lexicons, operational requirements documents (ORDs), operational concept descriptions, technical reference models, and doctrine. Applicable extracts from these sources may be used in the architecture itself as portions of architecture products. The completed architecture may then become an information source for other architecture development efforts.

Several architecture product types have been developed as part of this Framework guidance. Products of these types, if built in accordance with the guidance and examples provided, will allow architecture builders to capture the characteristics needed for particular analysis efforts. One important purpose of an architecture is to communicate among humans so that specific issues can be analyzed. In order to facilitate this human communication, most of the product types have been designed as graphics, supplemented by text as needed.

As listed in Section 3 and discussed in Section 4, the products form a continuum from the Operational Concept Diagram through the various products associated with operational, systems, and technical architectures. Generally, progressively more detail is provide in each successive product. Several products have been designed as overlays and extensions to basic product types; for example, cost and node information may be overlaid on activity models and the systems may be overlaid to node connectivity models. This increases the cohesiveness of the set, emphasizes linkages among component types, and maximizes reuse of individual products. Such reuse can occur, for example, when one architecture is used to address multiple issues: analysis of the issues may involve the same activity and data models, but require different overlays.

B-3 Interrelationships Among the Product Types

No matter what the purpose is for building an architecture, that architecture is intended to "tell a story" about a given subject area. In order for an architecture to tell a coherent story, all of its parts must be pieces of that same story, i.e., they must relate to each other in supplementary, not contradictory, ways. **Figure B-1** illustrates the relationships among the products. The set of



architecture products described here was designed with the intention of having the pieces operate synergetically, with each providing portions of the story that are built upon in turn by other products. For convenience, the products are categorized by the architecture type to which they most appropriately belong, in accordance with the definitions provided earlier. (In the figure, the categorizations are indicated by color-coding of the icons: blue for operational, orange for systems, red for technical, and purple for the core, which applies to all types.) In some cases, there are "gray areas," or cases in which a given product type seems to logically belong to more than one component. This is caused in part by the progressive nature of the products, which allows for "systems" characteristics to be overlaid onto "operational" diagrams. Another type of "gray area" is represented by the Data Dictionary and the Data Model which do not fit easily into any one product type category and are considered core products. The green arrows and text highlight the most important of these relationships.

The decision of what individual architecture products to build is made on the basis of the issue areas the architecture is intended to explore and the resulting characteristics that the architecture must capture and describe. A given architecture may therefore consist of all of the products or of a selected subset that is associated with one, two, or all three components. The "binning" of architecture products into operational, systems, or technical categories is a convenient way of describing them, but should not be considered a restriction in building an architecture.

B-4 How to Use the Framework

This section expands on the information provided in Section 3 in describing how to apply the Framework in building an architecture. A five-step procedure has been developed.

B-4.1 The Five-step Procedure

1. **Determine the intended architecture use:** In most cases, there will not be enough time, money, or resources to build purely top-down, all-inclusive architectures, even within a limited scope. Therefore, before beginning to describe an architecture, one must determine as specifically as possible the issues the architecture is intended to explore, the questions the architecture is expected to help answer, the resources available for building the architecture, and its expected audience and users. This focusing will make the effort more efficient and the architecture more usable.

2. **Determine the architecture scope, context, and any other assumptions to be considered:** Once the intended use has been decided, the prospective content of the architecture can begin to be addressed. Items to be considered include the scope of the architecture (functional, organizational, time-phased, etc.); the context (the architecture development effort's place in the larger scheme, scenarios, circumstances to be depicted); and any other assumptions such as the economic situation or the availability and capabilities of specific technologies at specific times in the future.

3. **Based on the intended use and the scope, determine which characteristics your architecture needs to capture:** Care should be taken to determine which characteristics of the subject area need to be described in order to satisfy the purpose of the architecture. If pertinent characteristics are omitted, the architecture may not be usable; if unnecessary characteristics are included, the architecture effort may prove infeasible given the time and resources available, and the architecture may never be built. Care should be taken as well to predict the future uses of the architecture so that, within resource limitations, the architecture can be structured to accommodate future tailoring and reuse.

4. **Based on the characteristics to be displayed, determine which architecture components/products should be built:** Depending on the outcomes of steps one through three, it may not be necessary to build the complete set of architecture components and products. In that case, only those products that display the required characteristics

should be built, and care should be taken to ensure that the products within that subset are consistent and properly related.

5. **Build products, use architecture:** The critical last step is to build the required set of architecture products and to use the architecture for the intended purpose. If the architecture needs some tailoring in order to serve its purpose, that tailoring should be done as efficiently as possible. In this regard, it may be useful, resources permitting, to conduct some proof-of-principle analysis of the architecture at various stages of completion.

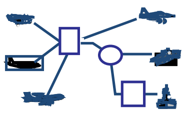
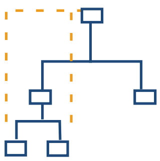
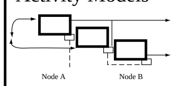
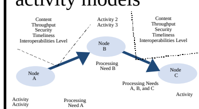
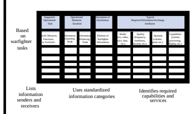
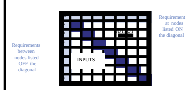
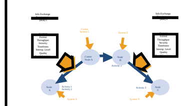
Table B-1 shows which of the steps described above promote each of the guiding principles described in Section B-1.

Table B-1: Architecture Development Steps Related to Guiding Principles

Principles / Steps	Architectures should be built with a purpose in mind.	Architectures should facilitate, not impede, communication among humans.	Architectures should be relatable, comparable, and integratable across DoD.	Architectures should be modular and reusable.	Short-term architecture efforts should be useful.
Determine the intended architecture use.	The intended use is the purpose.		Intended use will determine which architectures to compare.		Intended use will focus effort.
Determine scope, context, environment, other assumptions.		Scoping information will guide reader.	Scoping information will determine which architectures to compare.		Tailored scope will focus effort.
Determine which characteristics to capture.	Proper choice of characteristics will promote purpose.		Architectures can be compared IAW applicable characteristics.	Additional characteristics can be added for further analyses.	Saves time.
Determine which products should be built.	Product choice can emphasize purpose.	Eliminates unneeded and distracting products.		Products can be added for further analyses.	
Build products, use architecture.	Proper products & use satisfy purpose.	Proper products & use facilitate communication	Consistent products promote comparability, interface descriptions promote integration.	Consistent products promote modularity.	

B-5 Architecture Development Aids

Three architecture development aids have been designed that can facilitate the five-step procedure described.

	APPLICABLE ARCHITECTURAL COMPONENT					
	Operational	Operational	Operational	Operational	Operational	Operational
Architecture Products	High-level Operational Concept Graphic 	Command Relationships Charts 	High-Level or Generic Activity Model, Family of Related Activity Models  Overlays to Activit Model(s), e.g., arrows on activity models 	Information Exchange Matrix 	Required Capabilities Matrix 	Basic Node Connectivity Model 
Characteristics to Capture	High-level graphical description operational concept (high-level organizations, missions, geographic configuration, connectivity, etc.)	Command, control, coordination relationships among organizations	Multiple viewpoints (org./domain) on: scope of activities, activities, info exchange among activities, external info exchanges Activities assigned to nodes (organizations, facilities, workstations, etc., depending on level of architecture), activity-based cost information overlays	Supported Activities; consuming & producing nodes; information elements exchanged; attributes of the exchange (quality, frequency, speed, etc.)	Sum of required node-to-node capabilities (voice, video, text, security, etc.) Total requirements at specific nodes (processing, display, etc.)	Physical nodes; connectivities; activities performed at each node; information exchanged among nodes; throughput, format, timeliness, security, interoperability requirements, quality, required services per exchange

This matrix is designed to help organizations build architectures to answer specific questions or explore issue areas. It will help focus thinking and answer procedural questions such as "What architecture components do I need to build, and what specific model/representations do I need to build, using what methodologies or descriptive techniques?" The steps in using this matrix are described below.

- Determine intended architecture use (e.g., document capabilities, assess issues,...)
- Determine architecture scope, context, environment, and any other assumptions to be considered.
- Based on the intended use and the scope, determine which characteristics your architecture needs to capture.

Figure B-2: The Architecture Development Matrix

B-5.1 The Architecture Development Matrix

The Architecture Development Matrix summarizes the five-step procedure for developing an architecture, and provides some visual aids for following the procedure. The matrix provides lists of characteristics that may need to be captured in building a given architecture. When a needed characteristic is identified from the listings at the bottom of the matrix, the reader should look up from that listing to determine the graphic style of the product recommended for capturing that characteristic, the product's name, and the architecture component of which it is a part. The set of products thus shown to be needed for capturing the appropriate characteristics is the set of products that should be built. The Product Interrelationship Graphic can then help to determine the necessary relationships among the products. See **Figure B-2,** The Architecture Development Matrix.

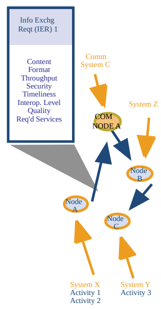
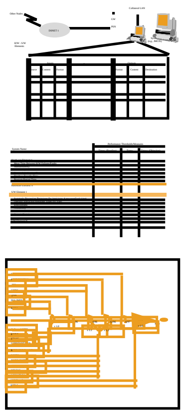
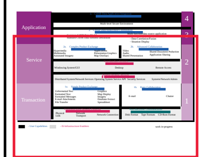
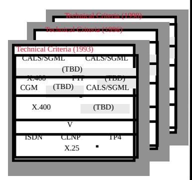
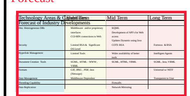
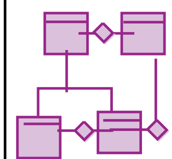
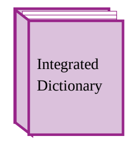
<table>
<tr><th colspan="7">APPLICABLE ARCHITECTURAL COMPONENT</th></tr>
<tr><td></td><td>Systems</td><td>Systems</td><td>Technical</td><td>Technical</td><td colspan="2">Core Information Products that Relate to All Other Products</td></tr>
<tr><td>Architecture Products</td>
<td>Overlays to Node Connectivity Model
</td>
<td>Supplements to Node Connectivity Model
</td>
<td>Standards, Criteria
Required Levels of IS Interoperability

Technical Criteria Profiles
</td>
<td>Technology Forecasts
Technology Domain: Data Production and Management
Forecast

Issues
• Hyperlink
• Profiling & Push
• Throttling
• Security</td>
<td>Attributed Information Model
</td>
<td>Integrated Dictionary of Architecture Terms
</td></tr>
<tr><td>Characteristics to Capture</td>
<td>Adds systems at each node, comm resources.</td>
<td>Info exchanges between systems, system elements and interfaces, applications and H/W allocated to system elements, performance parameters, evolution plans</td>
<td>Technical reference models, information criteria, time-phased criteria profiles (e.g., levels of interoperability), criteria deficiencies</td>
<td>Expected emerging technologies and specific products, timeframes, confidence factors, impacts on architectures from multiple viewpoints</td>
<td>Data elements, attributes, relationships</td>
<td>Definitions of all terms in representations</td></tr>
<tr><td colspan="7">• Based on the characteristics to be displayed, determine which architecture components/products should be built.
- One, two, or all three architecture components (operational, systems, technical) may be needed for a given purpose.
- For each component, one, two, or all of the constituent products may be required (also partial products).
- Ensure appropriate inter-product relationships are maintained. (See Product Interrelationships Graphic and Product Interrelationships N2 Chart.)
• Build products, use architecture.</td></tr>
</table>

Figure B-2: The Architecture Development Matrix (Cont)

B-5.2 The Product Interrelationships Graphic

The Product Interrelationships Graphic shows the major, high-level relationships among the architecture products and illustrates the general sequence (shown clockwise starting with the Technology Forecast, which is continuous) in which they would be built, assuming that all are required for a given architecture. Figure B-1, Product Interrelationships, illustrates that the Technology Forecast influences the thinking process in the construction of all the other products. The text items describe the "pieces of the story" that each product provides to others. The two "information products," the Data Dictionary and the Data Model, are shown inside the circle of products in order to show that they support and are fed by all the other products.

B-5.3 The Product Interrelationships N^2 Chart

The Product Interrelationships N^2 Chart provides more detail on the connections and relationships among the architecture products and can be used in deciding which products need to be built for a given architecture. By reading to the left and upward from each product type listed at the left of the chart, one can see what each product provides that other products build on. Keeping these kinds of product relationships in mind can help to assure that all products contribute to the same architecture "story," and that the products selected to be built for a given architecture forms a coherent and logically related set. See **Figure B-3**, Product Interrelationships N^2 Chart.

Individual products are described in more detail in Section 4.

Architecture Products	Operational Concept Diagram	Command Structure Charts	Activity Models	Overlays to Activity Models	Basic Node Connectivity Model	Attributed Information Model	Overlays to Node Connectivity Model	Supplements to Node Interoperability Model • System Element/ Interface Diagram	• System Performance Parameters Matrix	• System Evolution Diagram	Tailored Technical Criteria Profiles	Technology Forecast	Integrated Dictionary
Operational Concept Diagram		Basis for allocating command	Basis for building pool of activities	Broad categories of nodes	High-level nodes, connectivity needs	High-level systems	High-level terms			High-level “To-Be” concepts		Broad areas for technology investigation	Terms
Command Structure Charts	Suggests alternate concepts			Alternate structures to use as model viewpoints			High-level terms, relationships						Terms
Activity Models	Points out flaws in concepts	Points out flaws in command structures		Basis for overlays: activities, information exchange	Activities for allocation to nodes	Terms, attributes, relationships							Terms
Overlays to Activity Models	Points out flaws in concepts	Points out flaws in command structures	Points out flaws in activity models		Node activity paris	Terms, attributes							Terms
Basic Node Connectivity Model						Terms, attributes	Framework for assigning systems to nodes						Terms
Attributed Information Model			Helps to discover business rules that affect activities										Terms
Overlays to Node Connectivity Model						Terms, attributes		Node/system structure for further decomposition					Terms
Supplements to Node Connectivity Model: • System Element Interface Diagram						Terms, implied attributes			System-to-system data flow framework for performance requirements				Terms
• System Performance Parameters Matrix						Terms, implied attributes				Requirements basis for migration decisions			Terms
• System Evolution Diagram						Terms, implied attributes					System plans, scope for criteria profiles		Terms
Tailored Technical Criteria Profiles			Applicable technical criteria for detailed analysis		Definitions of levels of interoperability	Terms, implied attributes incl. levels of interop.	Time-phased governing criteria, incl. levels of interop.	Time-phased governing criteria, incl. levels of interop.	Time-phased governing criteria	Time-phased governing criteria			Terms
Technology Forecast	Technology predictions for concept development		New activities based on new technology	Technology implications on cost	New nodes resulting from new technology	New terms, attributes, relationships	New systems, comm. resources	New applications, H/W	New performance requirements	Technology predictions for realistic migration plans			Terms
Integrated Dictionary	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitons of terms	Definitions of terms	Definitions of terms	

Figure B-3 Product Interrelationships N² Chart

This Page Intentionally Left Blank

ANNEX C
The Broadcast/Receive Example Architecture Study

C.1 Background

An example case study has been undertaken that illustrates both the principles of the Framework guidance and an applicable set of architecture products. The example selected is the Broadcast/Receive Working Group effort that was jointly sponsored by CISA and the Intelligence Systems Secretariat/Intelligence Systems Board (ISS/ISB).

As a part of the larger intelligence migration initiative to reduce the number of separate intelligence systems, the working group was formed in 1994 to examine the existing Ultra High Frequency (UHF) intelligence broadcast services and one emerging concept for combining the services (Binocular). The group's objective was to uncover and examine issues surrounding the potential combining of service functionality into a smaller number of services–functional redundancy, inefficiency, and impacts on the user; resource duplication; formatting issues; bandwidth contention–and to make recommendations concerning concepts warranting further study. It is important for purposes of this illustrative case study to realize that the purpose of the working group was not primarily to *solve* issues, but rather to identify, frame, and examine them and to recommend next steps.

The part of the group's work that is pertinent to the Framework guidance is the modeling effort that was undertaken to provide a basis for quickly comparing the different services. The primary method used to provide this comparison was activity modeling: a template model was constructed that provided a view of the activities any intelligence broadcast service would perform; then, a separate model was constructed of each service, and the Binocular concept, following the template's format but tailored to describe the individual activities and characteristics of each service. A corresponding node diagram was also prepared for each service, which provided a more physical view. This means of comparatively evaluating the broadcast services facilitated communication within the group, revealed some issues, and focused the group's recommendations on issue areas.

The case study presented here is in a sense a "reverse engineering" of the group's work. It describes how the work could have been done in accordance with the Framework guidance, if the guidance had been available then. That is, the case walks through the Framework's five recommended steps to building an architecture and provides examples of the appropriate architecture products. Some of the products shown were actually developed during the working group, some are modifications of actual products, and some are newly created for the case study. *The intent of these modified and new products is not necessarily to construct complete products that are accurate in every detail. Rather, the intent is to provide enough plausible detail to be able to illustrate the recommended process, the applicable products, and the linkages among the products; and to show how these products help to realize the objectives of the architecture.*

C.2 AN ARCHITECTURAL THREAD: *The Development and Integration of Architecture Products That Support Intelligence Broadcast Issues Analysis*

Steps

1. **Determine intended architecture use (e.g., document capabilities, assess issues)**

   The Intelligence Broadcast architecture will be used to compare and assess capabilities of similar intelligence broadcast services, in identifying potential issues for system migration and recommending avenues for resolving those issues.

2. **Determine architecture scope, context, environment, and any assumptions to be considered**.

   Much of the scope and context from the original study also applies to this architectural thread:

   - Focus on UHF intelligence broadcast services, namely Tactical-Related Applications/ TRAP Data Dissemination System (TRAP/TDDS), Tactical Intelligence Broadcast Service (TIBS), Tactical Reconnaissance Intelligence Exchange System (TRIXS), and the Binocular Concept.
   - "Service" refers to an entire Broadcast/Receive provider/ receiver/user (i.e., TRAP/TDDS, TIBS, TRIXS, Binocular); "system" refers to automated information systems, communications systems, sensors, receivers, etc., used in conjunction with or as a part of the broadcast service.
   - Focus on information flow/exchange.

Specifically for purposes of this Architectural Thread the focus has been narrowed down:

- Scope problem-solving down to one issue (the "Common Format" issue) from an identifiable set of issues. This represents additional narrowing of scope for this example thread.
- Maintain development of the architecture at an unclassified level. Since this study is not intended to be a full-blown architecture, but rather simply a thread, the classified information contained in the referenced report to the Intelligence Broadcast Working Group has been omitted from this case study.
- Focus on comparison of broadcast capabilities, independent of programmatics, in issue identification and problem-solving.

3. **Based on the intended use and the scope, determine which characteristics your architecture needs to capture**.

Based on the intended use, scope, and assumptions from steps one and two, this architecture thread needs to show:

- A high-level functional description of intelligence broadcasting
- Activities that are supported by one or more of the services
- Key nodes (receive or transmit) that support each of the broadcast services
- Activities that each node performs
- Systems used by each node
- Possible issues related to activities
- In-depth view of additional formatting requirements and attributes
- Issue-specific detail (i.e., formatting requirements, attributes, and information exchange)
- Definitions of terms that will facilitate a common understanding.

4. **Based on the characteristics to be displayed, determine which architecture components/products should be built.**

Using the Architecture Development Matrix as an aid, one can determine that the following products need to be built; the rationale for each product is shown below in Figure C-1.

Figure C-1: Products Required for Example Architecture Thread

Characteristic Needed	Why Characteristic Needs to be Captured	Product to Build to Capture Characteristic
High-level activities	Get disparate group reading off "same sheet of music"	Generic activity model
Activities supported by one or more of the services.	1-page comparison of functional scope	Color-coded hierarchy chart
Information exchanges	Comparison of services information exchanges, functional complexity	Activity models of each service
Definitions of terms used	Get disparate group reading off same sheet of music	Integrated dictionary
Key nodes (XMIT/RCV) that support the services	Facilitate comparison of nodes' functional redundancy	Basic node connectivity model of each service
Activities keyed to nodes	Facilitate comparison of nodes' functional redundancy	Basic node connectivity model of each service
Systems used by nodes	Examine system redundancies	Systems overlays to node connectivity models
Possible issues related to activities	Frame issues for selection (format issue selected)	Overlays to activity model (possible issue areas highlighted)
In-depth view of additional formatting requirements and attributes	Issue selected may require more depth	Overlays to activity model (annotations on arrows)
Detail of services' formatting processes	Issue selected does require more depth	Overlays to activity models (decomposition)
Further detail on information exchanges as appropriate for issue	Need to illustrate issue for group discussion	Overlays to activity models (further decomposition)

5. **Build products, use architecture**.

The products built are illustrated below (**Figures C-2** through **C-10**) with explanatory text showing how they helped in comparing the services, in scoping down the architecture thread to one issue, and in illustrating the issue. Figure subtitles are used to map the products listed in Figure C-1 to the product illustrations. In addition, the Product Interrelationships N2 Chart (**Figure C-11**) highlights the relationships between products built for this architectural thread.

One of the most important steps in architecture development is providing an integrated dictionary of terms to facilitate a common understanding of all broadcast activities and related terms of reference used in the architecture. The integrated dictionary has not been attached here, but would be when developing a full-blown architecture.

The modeling and the discussions it triggered resulted in a Working Group recommendation to undertake a formal study of the feasibility of using a common message format for all intelligence broadcast services.

A High-Level Graphic of the Operational Concept was not needed, because the Working Group was familiar with the concept. What was needed instead was a functional view for comparison purposes. ***A generic activity model of intelligence broadcasting was developed, based on an existing model of overall intelligence activities.***

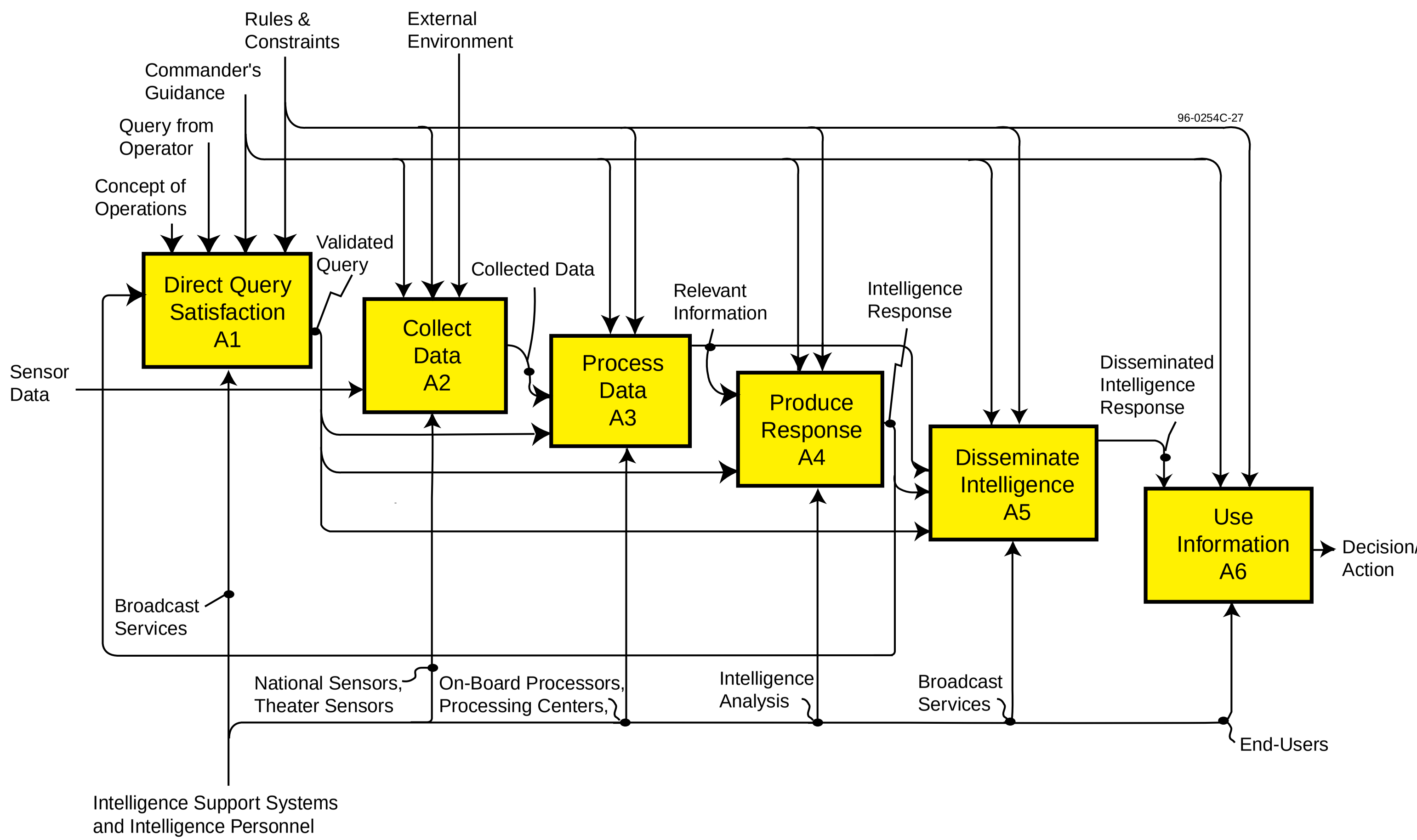


Figure C-2: High-Level Operational Activity Concept for Functions Supporting Intelligence Broadcasting (Generic Activity Model)

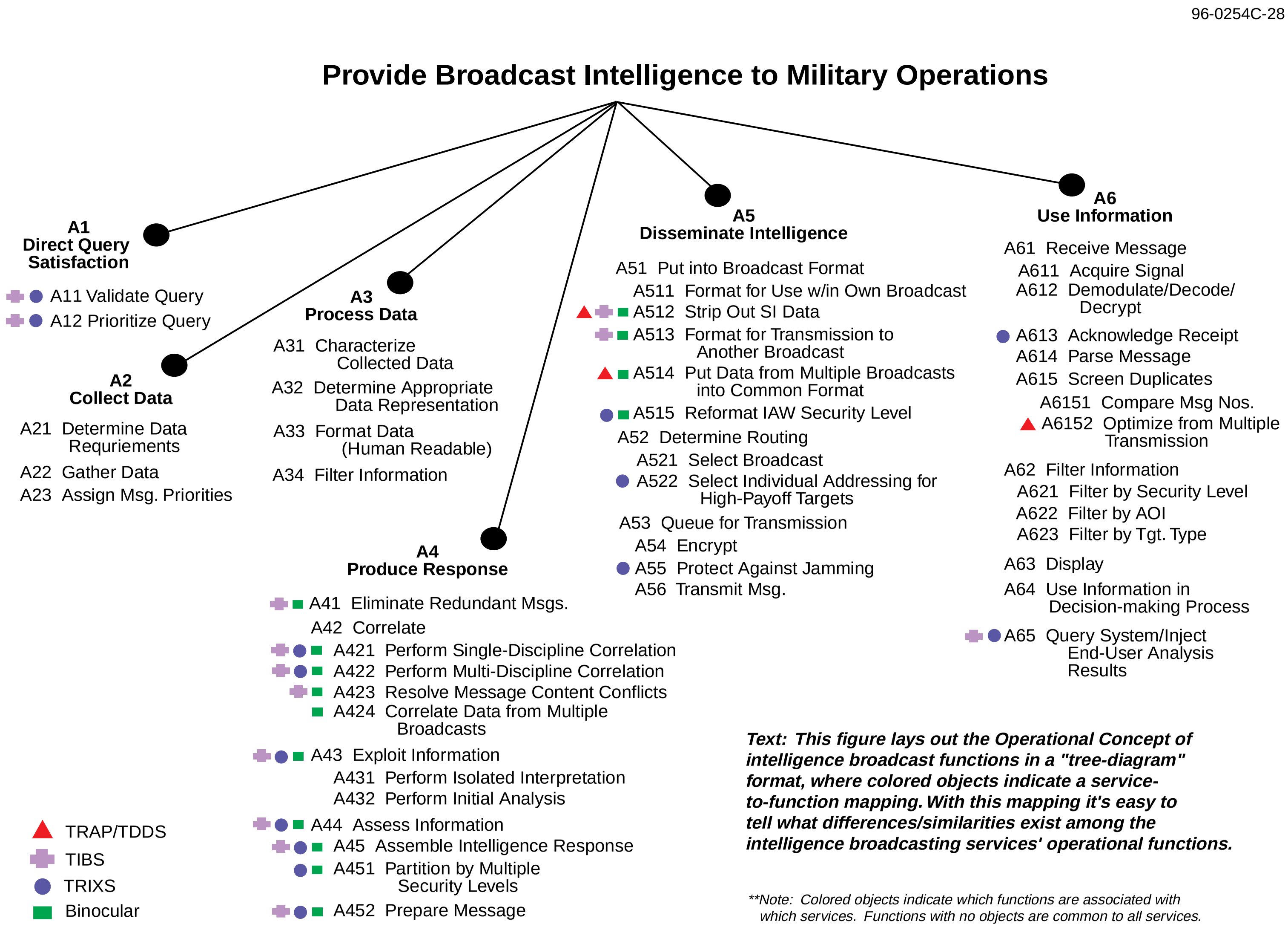
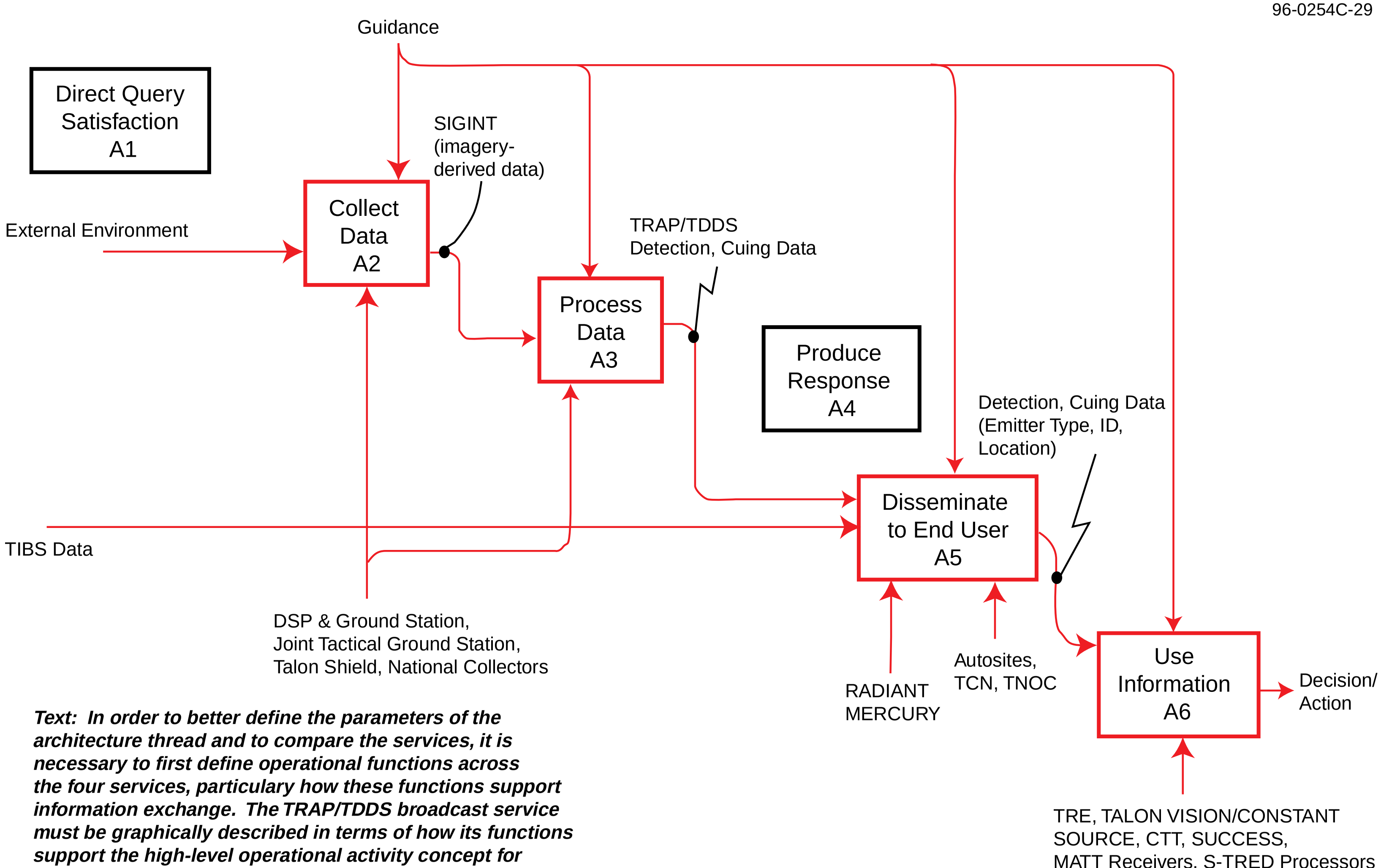


Figure C-3: Illustration of Overlap in Functionality of Services (Color-Coded Hierarchy Chart)

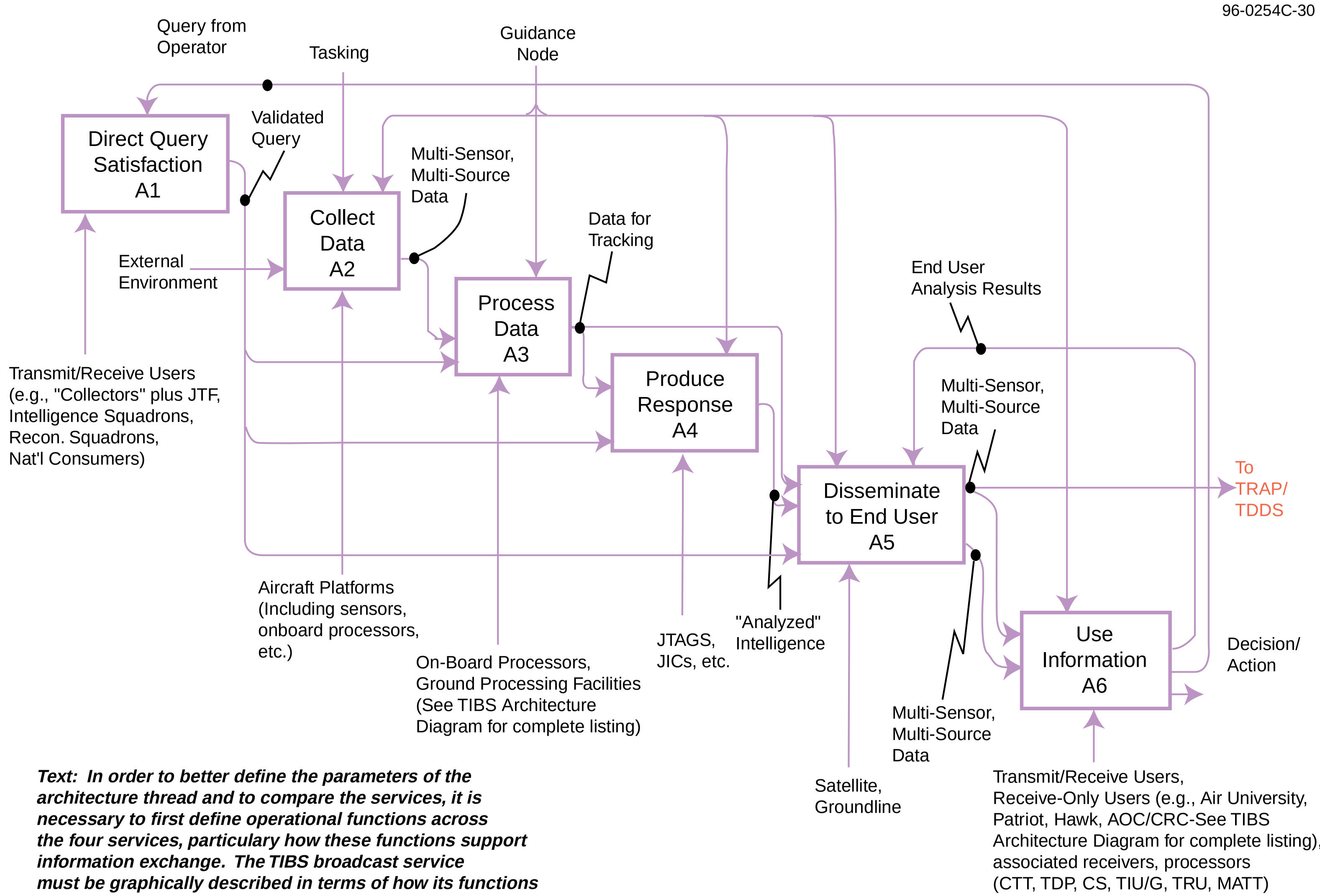
96-0254C-29



Text: In order to better define the parameters of the architecture thread and to compare the services, it is necessary to first define operational functions across the four services, particulary how these functions support information exchange. The TRAP/TDDS broadcast service must be graphically described in terms of how its functions support the high-level operational activity concept for intelligence broadcasting.

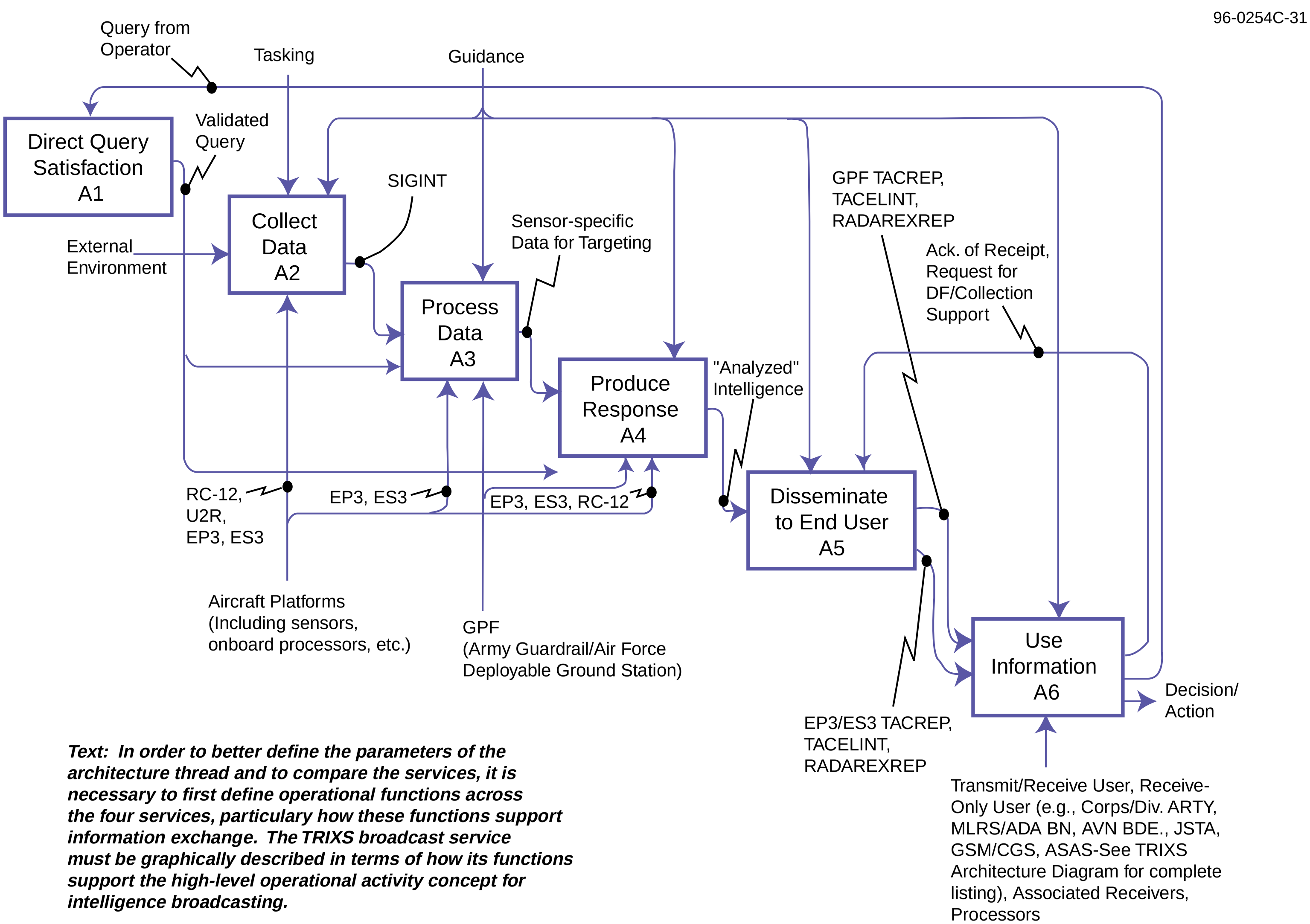
Figure C-4a: TRAP/TDDS Basic Functional Description (Activity Model)

96-0254C-30



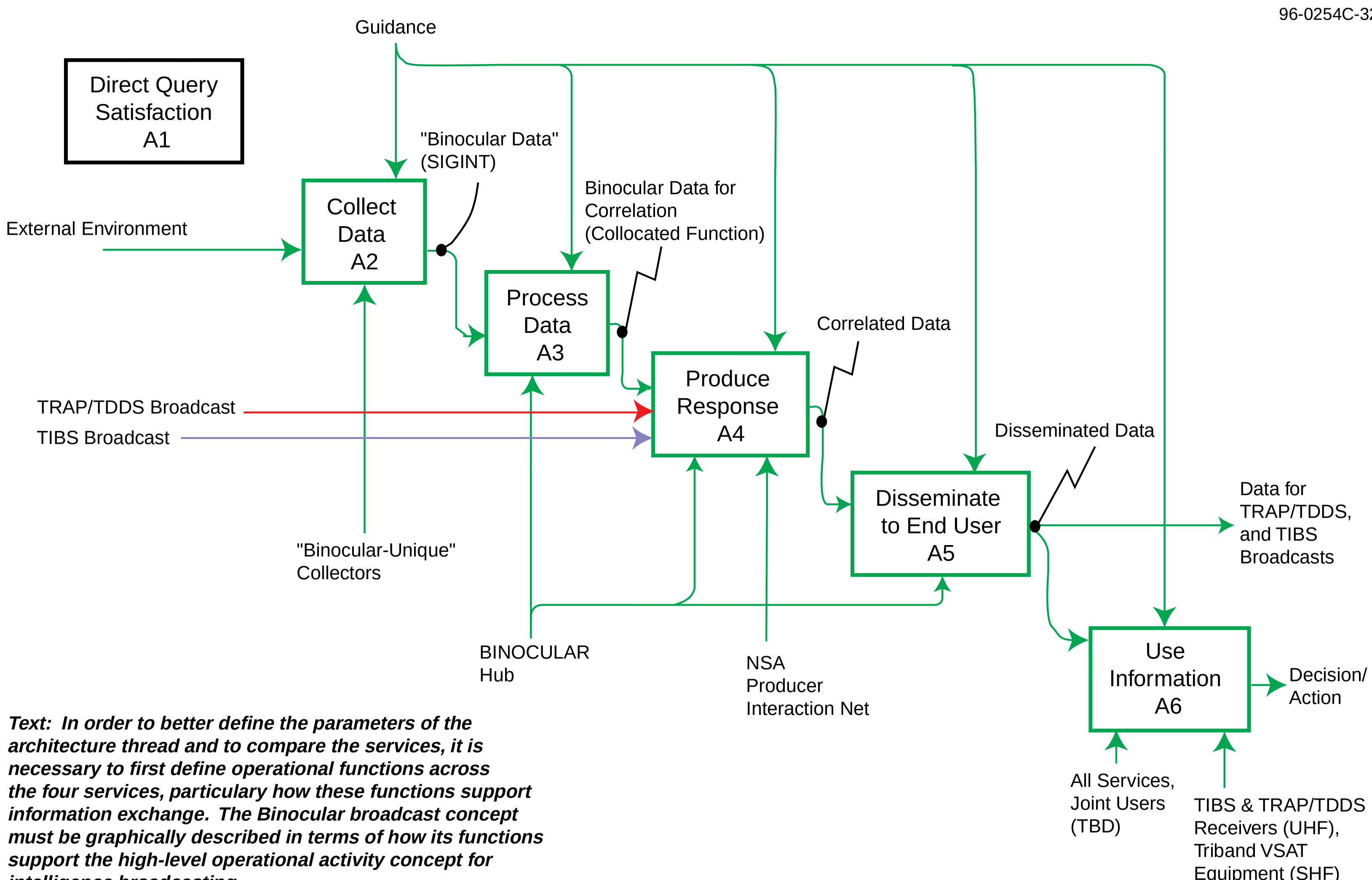
Text: In order to better define the parameters of the architecture thread and to compare the services, it is necessary to first define operational functions across the four services, particulary how these functions support information exchange. The TIBS broadcast service must be graphically described in terms of how its functions support the high-level operational activity concept for intelligence broadcasting.

Figure C-4b: TIBS Basic Functional Description (Activity Model)



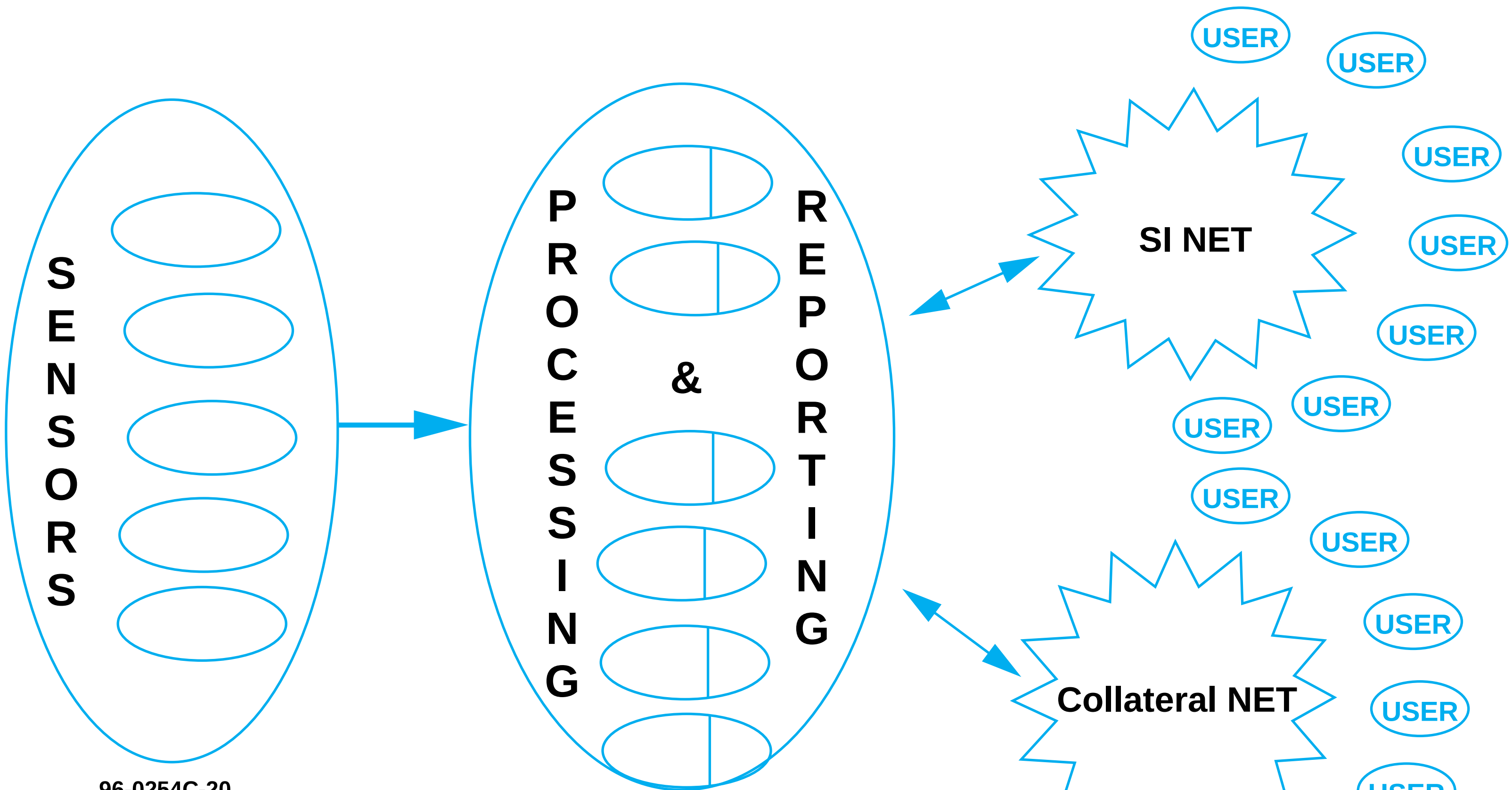
Text: In order to better define the parameters of the architecture thread and to compare the services, it is necessary to first define operational functions across the four services, particulary how these functions support information exchange. The TRIXS broadcast service must be graphically described in terms of how its functions support the high-level operational activity concept for intelligence broadcasting.

Figure C-4c: TRIXS Basic Functional Description (Activity Model)



Text: In order to better define the parameters of the architecture thread and to compare the services, it is necessary to first define operational functions across the four services, particulary how these functions support information exchange. The Binocular broadcast concept must be graphically described in terms of how its functions support the high-level operational activity concept for intelligence broadcasting.

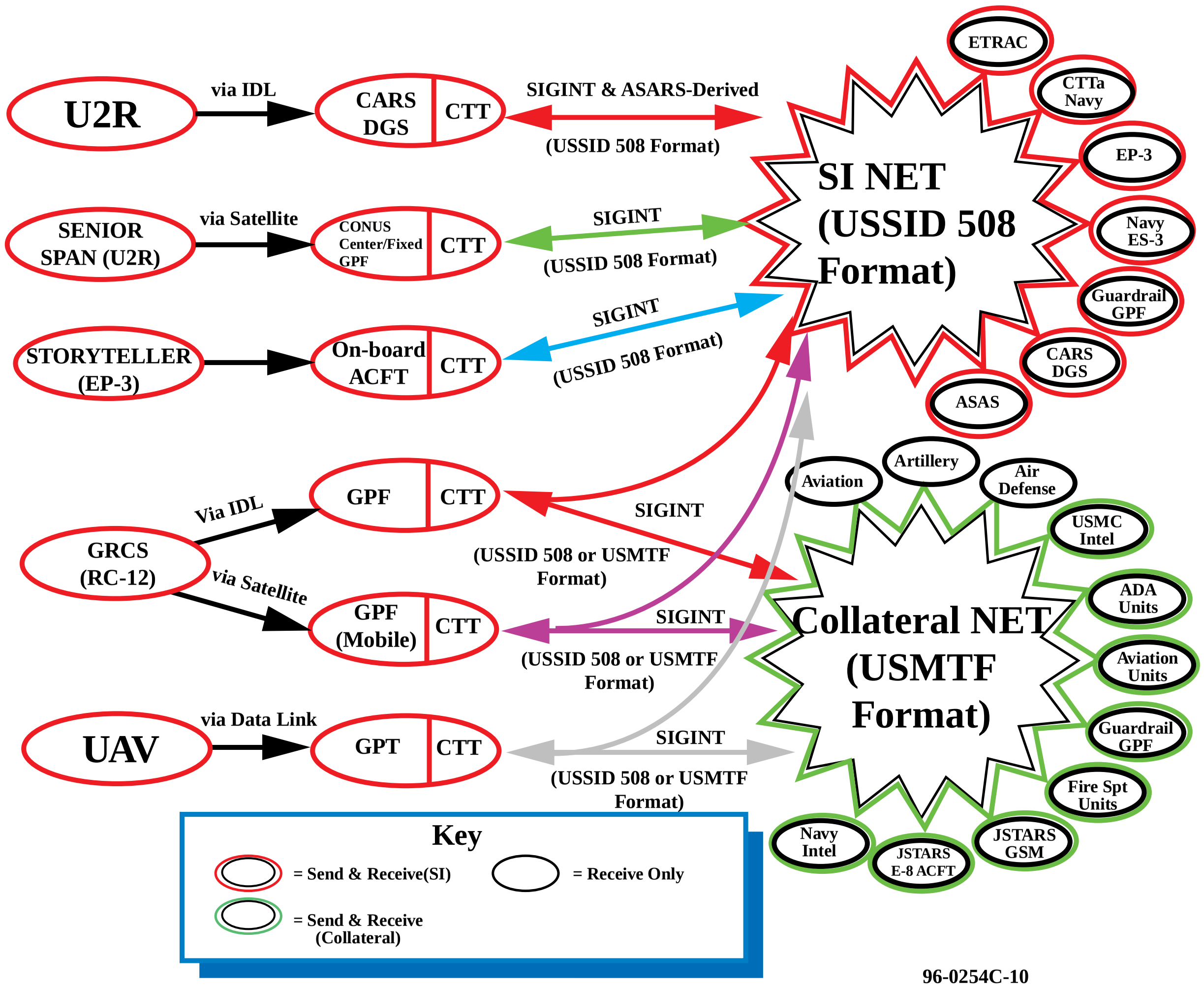
Figure C-4d: Binocular Basic Functional Description (Activity Model)



96-0254C-20

Text: It is important to identify and illustrate key nodes (receive or transmit) supported by each of the broadcast services and overlay those nodes with the activities that make up the intelligence broadcast concept. The following general nodal picture shows an unclassified slice of all four services in the example of the TRIXS-Related Operational Nodes.

Figure C-5: TRIXS-Related Operational Nodels (Basic Node Connnectivity Model)



Text: Once the operational nodes have been depicted, the system components of the broad cast service(s) are tied to nodes that the service(s) support.

Figure C-6: TRIXS-Related Systems Overlay to Operational Nodes (Systems Overlay to Node Connectivity Model)

96-0254C-33

***Text:** To scope the architecture thread's problem-solving agenda down to one issue, a set of issues must first be identified. After analyzing the differences in functionality the services shown in Figures 5-8A through 5-8D, we used the TRAP/TDDS functional diagram to illustrate a potential set of issues (in this case, three issues were identified) from which any single issue may be chosen for further analysis. From this set of issues, one issue is chosen for this sample thread. The "common format" issue was chosen based on its relevance in a follow-on study resulting from the modeling effort done on intelligence broadcasting. This diagram graphically depicts the issue set and its relevance to operational functions of the services.*

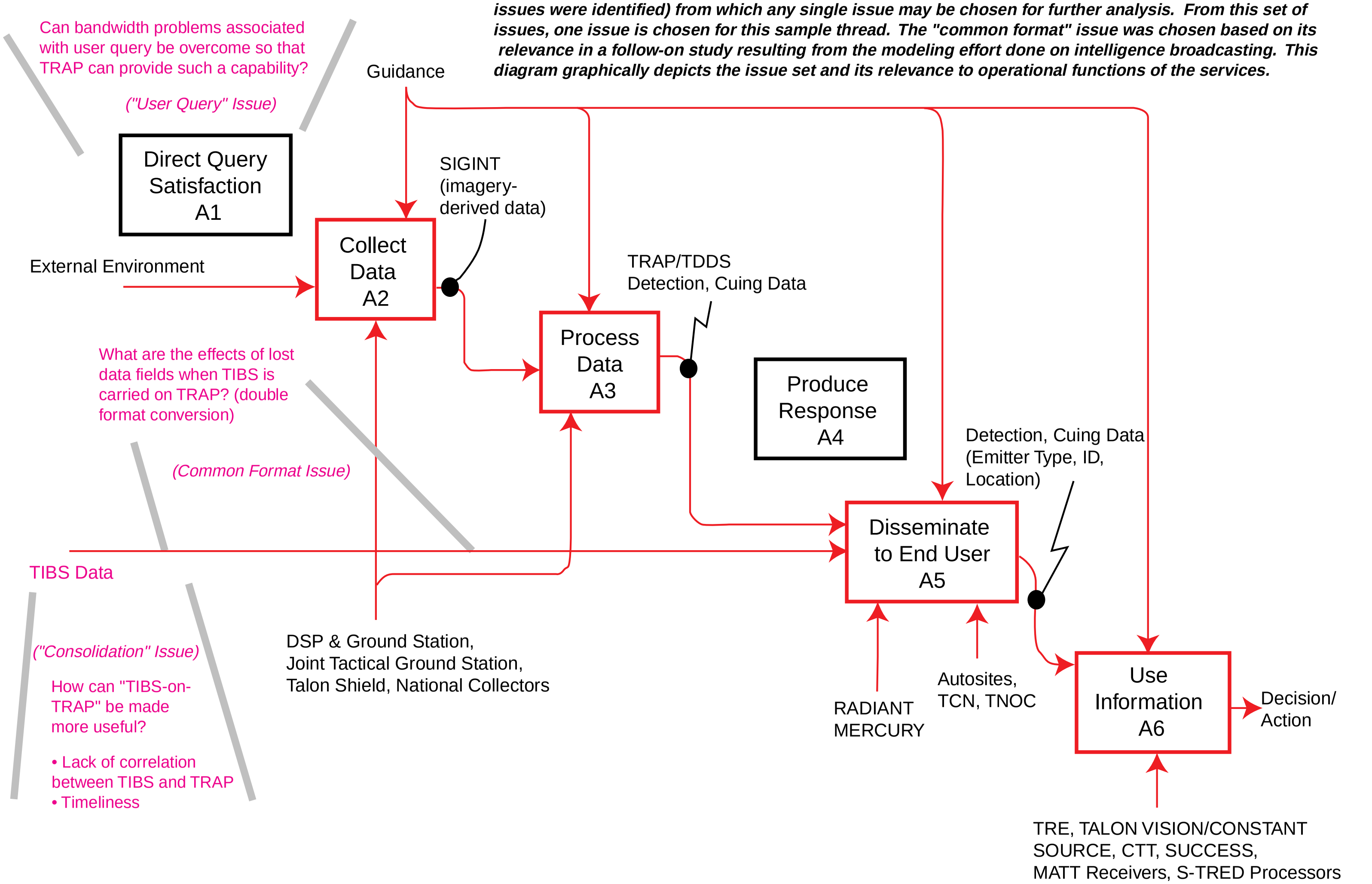
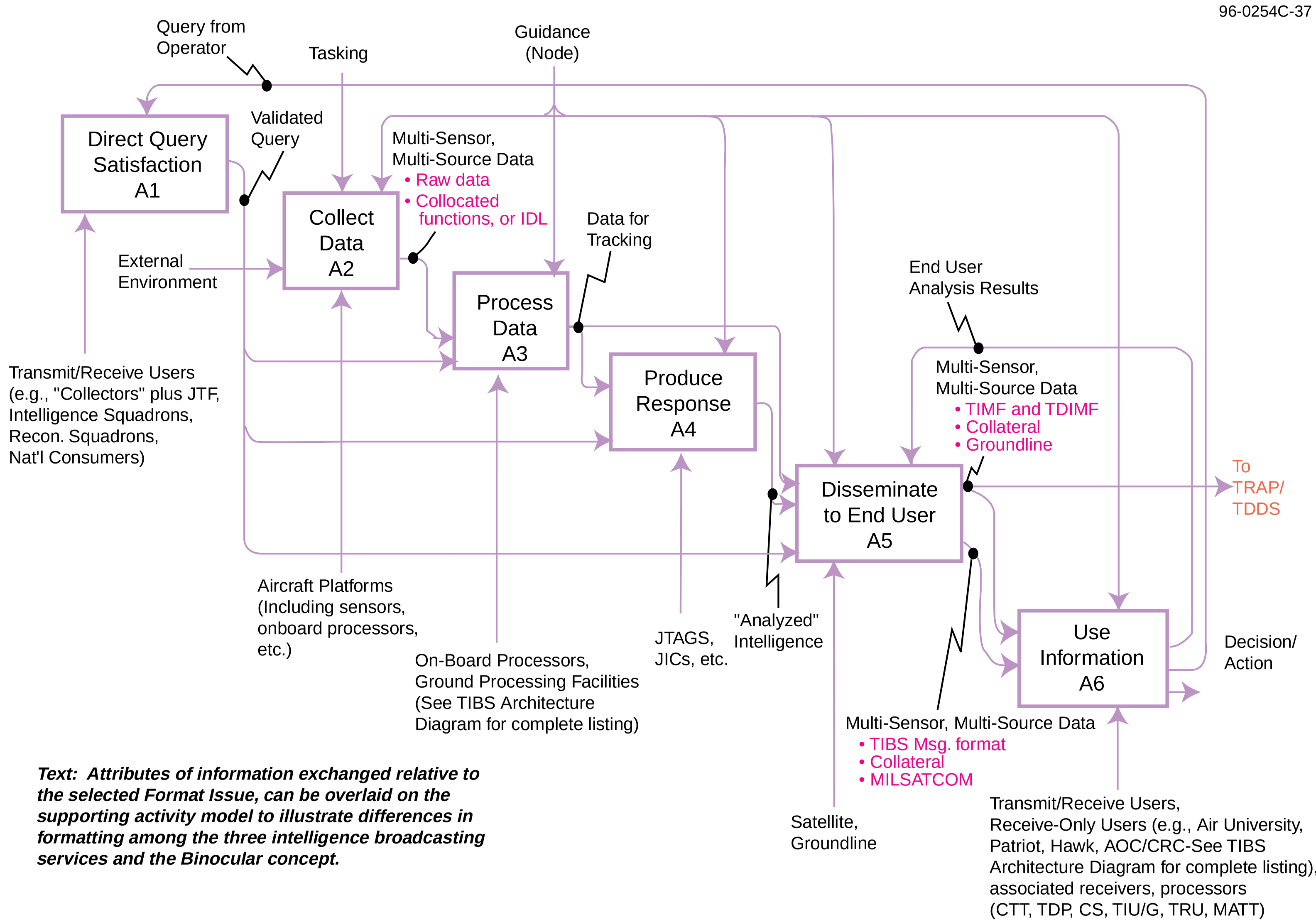


Figure C-7: TRAP/TDDS Example of Issue Identification (Overlay to Activity Model)



Text: Attributes of information exchanged relative to the selected Format Issue, can be overlaid on the supporting activity model to illustrate differences in formatting among the three intelligence broadcasting services and the Binocular concept.

Figure C-8a: TRAP/TDDS Annotated Overlay to the Functional Description (Overlay to Activity Model)

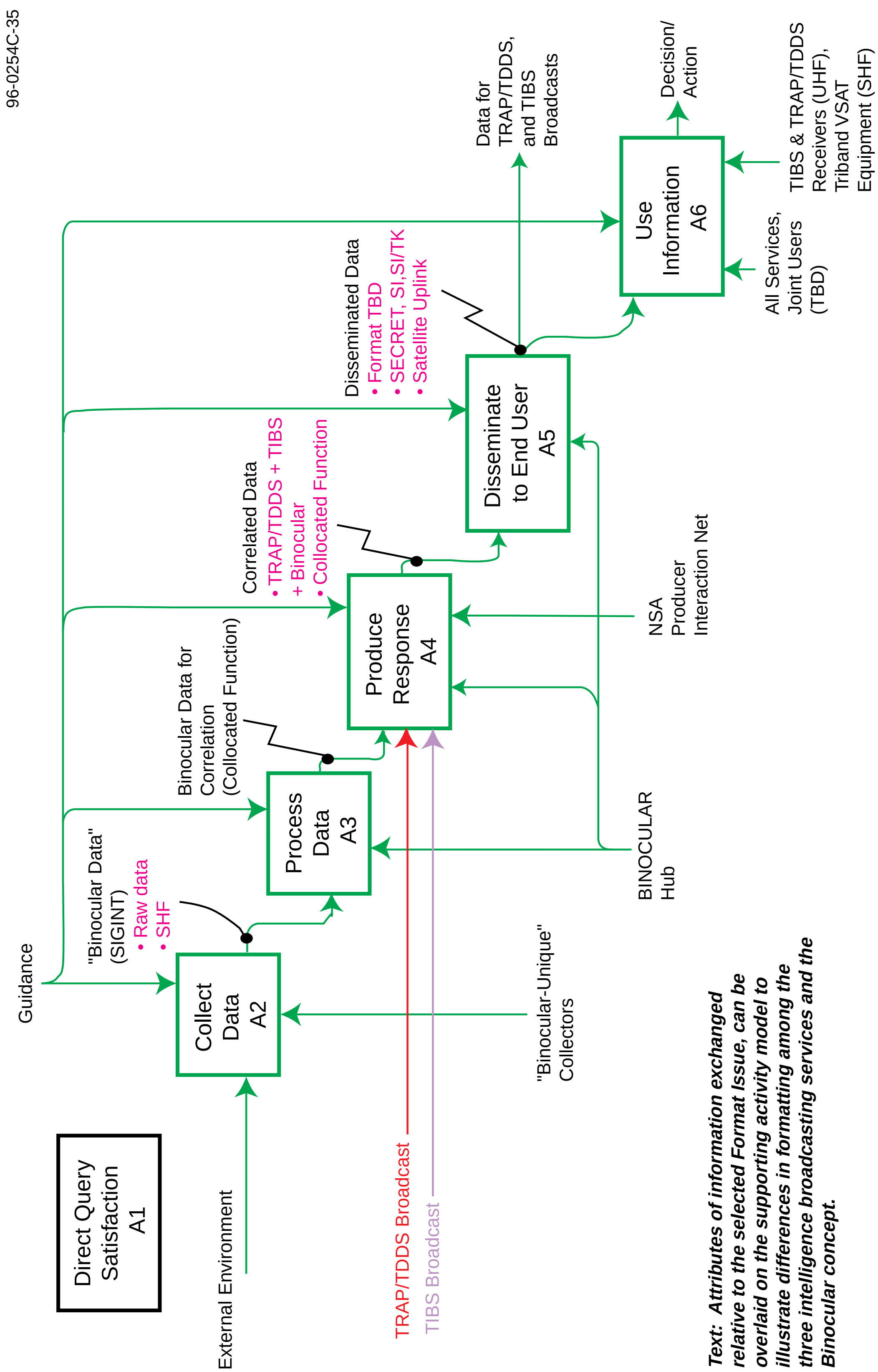
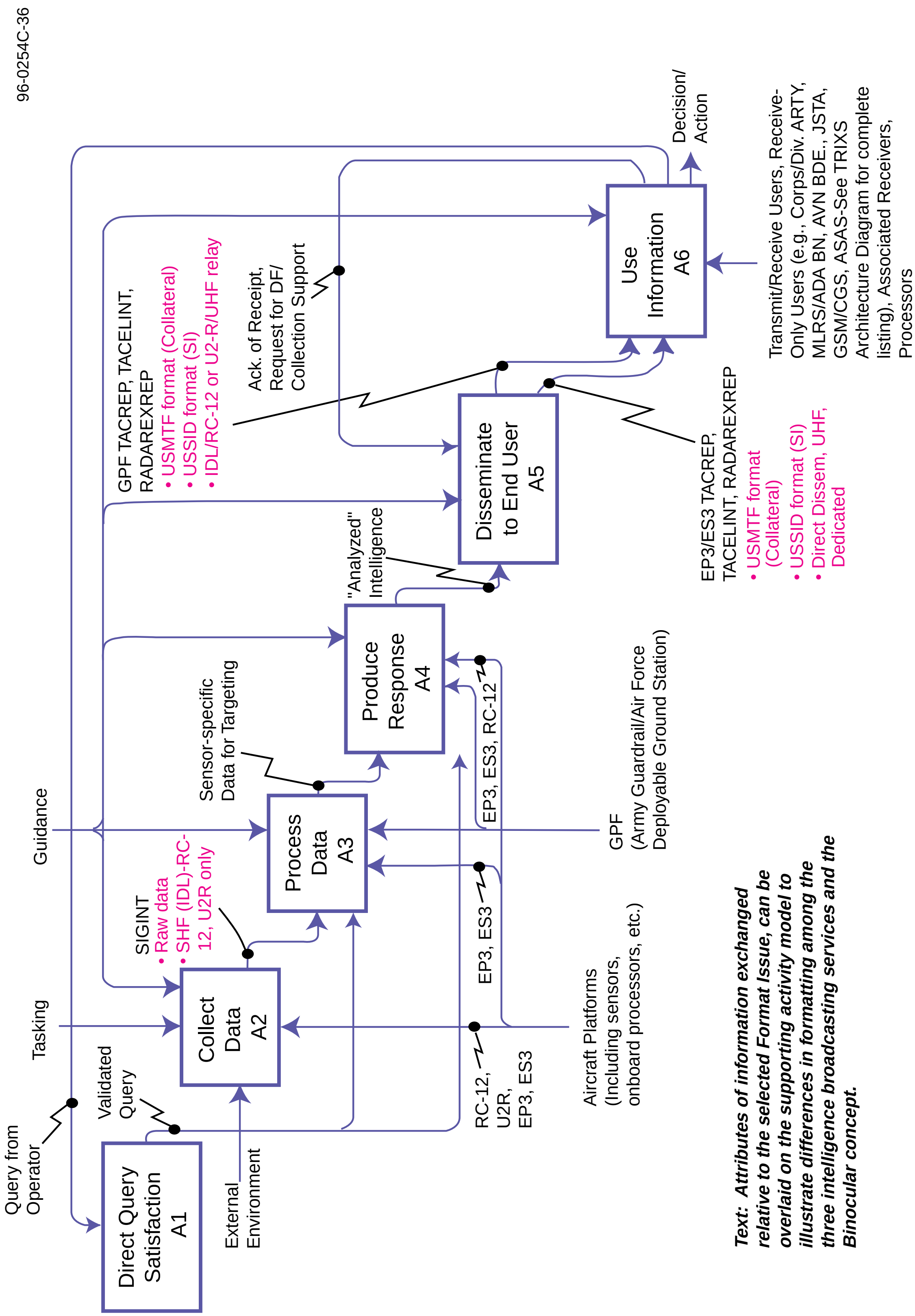


Figure C-8b: TIBS Annotated Overlay to the Functional Description (Overlay to Activity Model)



Text: Attributes of information exchanged relative to the selected Format Issue, can be overlaid on the supporting activity model to illustrate difference in formatting among the three intelligence broadcasting services and the Binocular concept.

Figure C-8c: TRIXS Annotated Overlay to the Functional Description (Overlay to Activity Model)

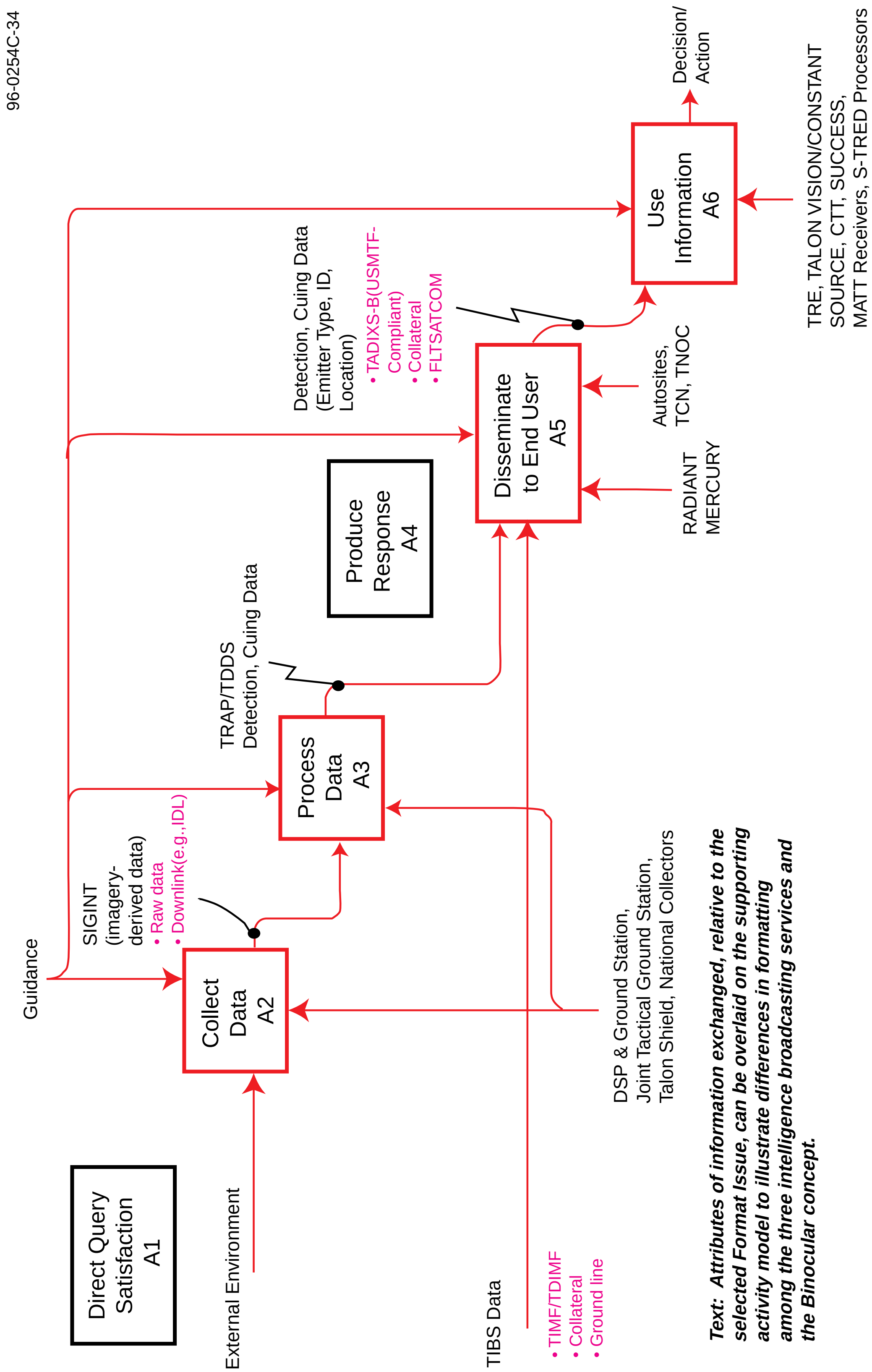
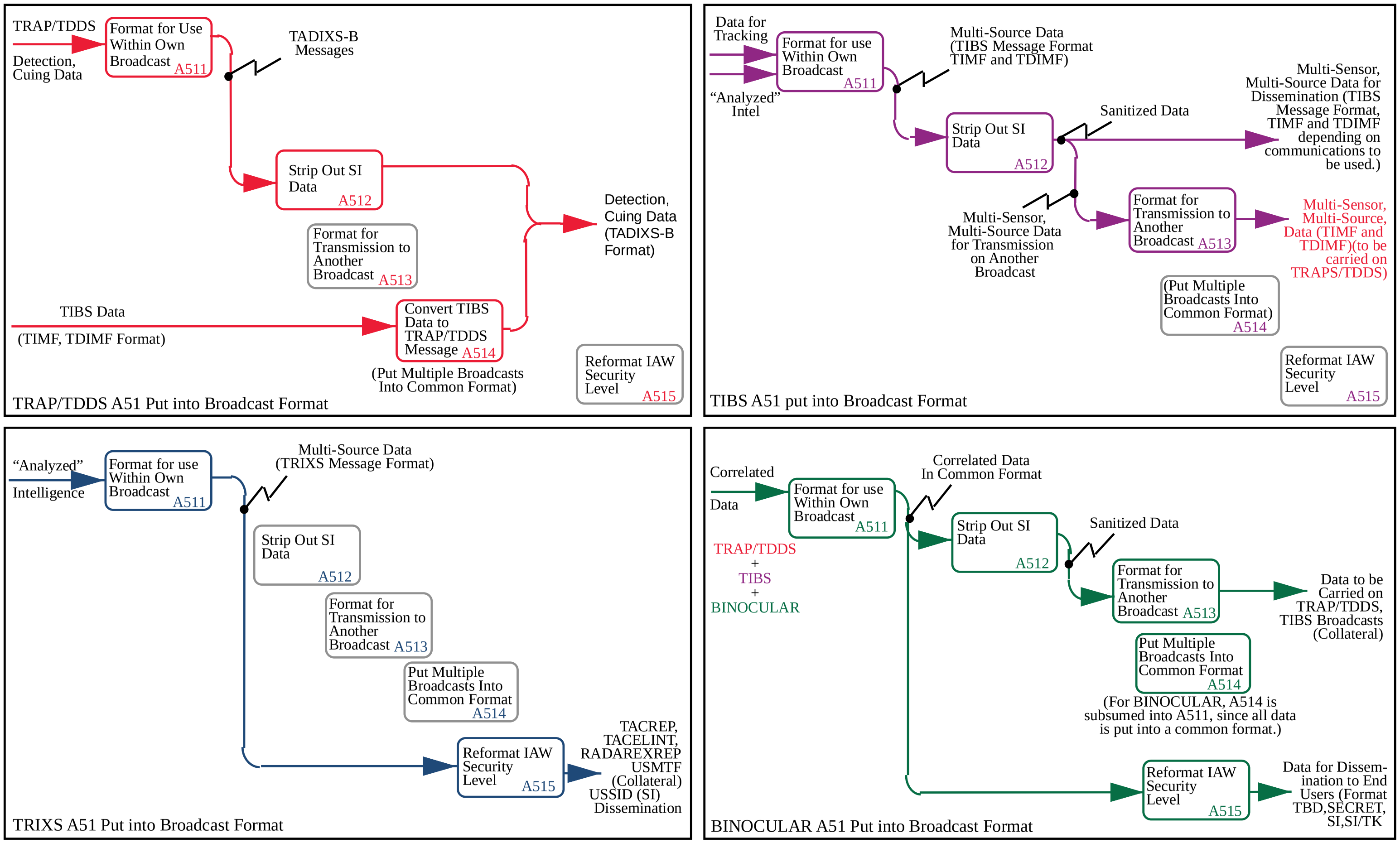


Figure C-8d: Binocular Annotated Overlay to the Functional Description (Overlay to Activity Model)

Text: The formatting issue can be examined by further decomposing the activity model to a level that allows a meaningful comparison of the services' formatting and reformatting requirements. The evolution of the attributes of the information as it progresses through the services' respective formatting functions can then be traced and compared.



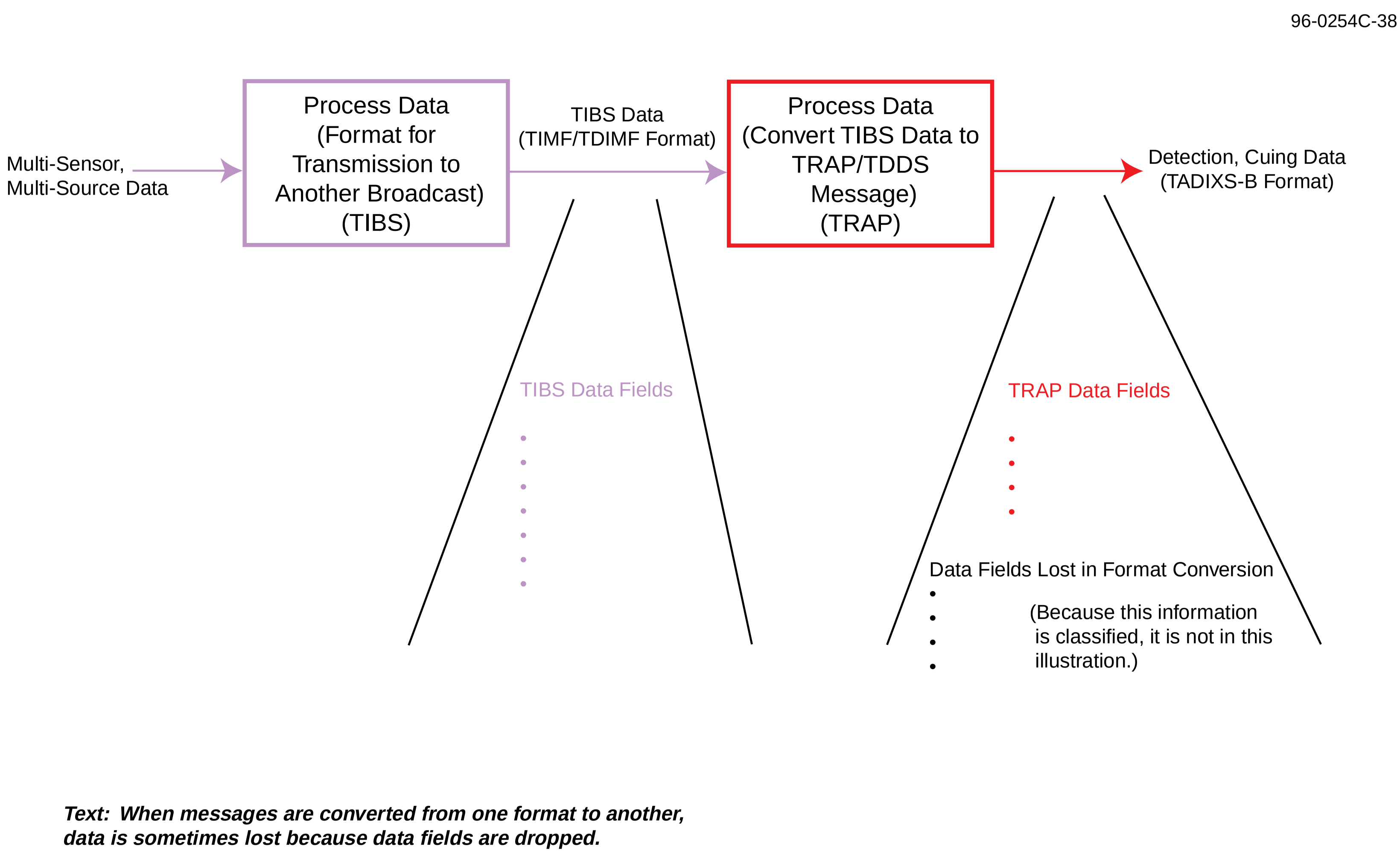


Figure C-10: Message Format Dropped Data Fields
(Overlay to Activity Model (Further Decomposition))

Architecture Products	Operational Concept Diagram	Command Structure Charts	Activity Models	Overlays to Activity Models	Basic Node Connectivity Model	Attributed Information Model	Overlays to Node Connectivity Model	Supplements to Node Interoperability Model • System Element/Interface Diagram	• System Performance Parameters Matrix	• System Evolution Diagram	Tailored Technical Criteria Profiles	Technology Forecast	Integrated Dictionary
Operational Concept Diagram		Basis for allocating command	Basis for building pool of activities	Broad categories of nodes	High-level nodes, con-nectivity needs	High-level terms	High-level systems			High-level "To-Be" concepts		Broad areas for technology investigation	Terms
Command Structure Charts	Suggests alternate concepts		Alternate structures to use as model viewpoints			High-level terms, relationships							Terms
Activity Models	Points out flaws in concepts	Points out flaws in command structures		Basis for overlays: activities, information exchange	Activities for allocation to nodes	Terms, attributes, relationships							Terms
Overlays to Activity Models	Points out flaws in concepts	Points out flaws in command structures	Points out flaws in activity models		Node-activity pairs	Terms, attributes							Terms
Basic Node Connectivity Model						Terms, attributes	Framework for assigning systems to nodes						Terms
Attributed Information Model			Helps to discover business rules that affect activities										Terms
Overlays to Node Connectivity Model						Terms, attributes		Node/system structure for further decomposition					Terms
Supplements to Node Connectivity Model: • System Element/ Interface Diagram						Terms, implied attributes			System-to-system data flow framework for performance rqmts.				Terms
• System Performance Parameters Matrix						Terms, implied attributes				Requirements basis for migration decisions			Terms
• System Evolution Diagram						Terms, implied attributes					System plans, scope for criteria profiles		Terms
Tailored Technical Criteria Profiles			Applicable technical criteria for detailed analysis		Definintions of levels of interoperability	Terms, implied attributes, incl. levels of interop.	Time-phased governing criteria, incl. levels of interop.	Time-phased governing criteria, including levels of interoperability	Time-phased governing criteria	Time-phased governing criteria			Terms
Technology Forecast	Technology predictions for concept development		New activ-ities based on new technology	Technology implications on cost	New nodes resulting from new technology	New terms, attributes, relationships	New systems, comm. resources	New applications, H/W	New performance requirements	Technology pre-dictions for realistic migration plans			Terms
Integrated Dictionary	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	Definitions of terms	

The highlighted cells indicate relationships between products built as part of the set for the architecture thread. This chart can be used to make sure that the highlighted relationships exist.

Figure C-11: Product Interrelationships N2 Chart

ANNEX D

ELEMENTS OF WARFIGHTER INFORMATION

To date, there is no community accepted taxonomy of warfighter information, i.e. that information required by the warfighter to accomplish his missions. The following is presented as a strawman and a potential starting point for developing such a list. **Figure D-1** graphically depicts four categories of warfighter information that form the highest level categorization in the taxonomy. **Table D-1** extends the concept represented in Figure D-1 and provides an initial list of standardized categories of warfighter information under which most elements of warfighter information can logically be placed. As stated above, this list is presented as a strawman and as a potential starting point for developing a Universal List of Warfighter Information, analogous to the Universal Joint Task List, that all unified commands, Services, and DoD agencies can use to describe the information categories and elements that are the subject of C4ISR information exchanges.

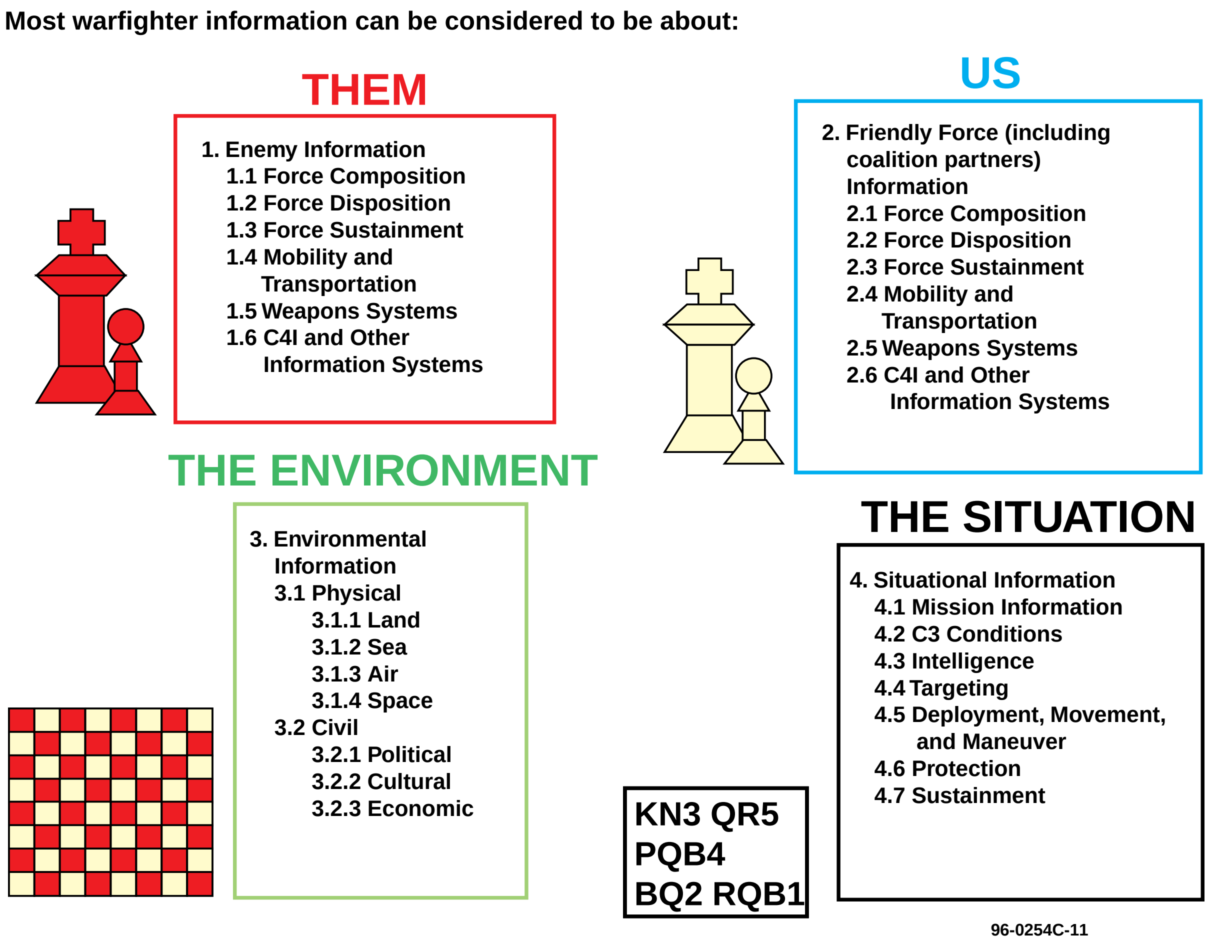


Figure D-1: Warfighter Information

TABLE D-1 STANDARD WARFIGHTER INFORMATION		
Information Type	**Information Category**	**Definition**
1. Friendly Force (including coalition partners) Information	1.1 Force Composition	Types and numbers of friendly military forces. Includes information on personnel.
	1.2 Force Disposition	Locations of friendly military forces.
	1.3 Force Sustainment	Logistics support (supply, maintenance, medical, etc.) capabilities.
	1.4 Mobility and Transportation	Capability for inter- and intra-theater movement of forces and materiel.
	1.5 Weapons Systems	Type, number, capabilities, and limitations of friendly weapon systems.
	1.6 C4I and Other Information Systems	Type, number, capabilities, and limitations of friendly C4I and other information processing systems.
2. Enemy Information	2.1 Force Composition	Types and numbers of enemy military forces.
	2.2 Force Disposition	Locations of enemy military forces.
	2.3 Force Sustainment	Logistics support (supply, maintenance, medical, etc.) capabilities.
	2.4 Mobility and Transportation	Capability for inter- and intra-theater movement of forces and materiel.
	2.5 Weapons Systems	Type, number, capabilities, and limitations of enemy weapon systems.
	2.6 C4I and Other Information Systems	Type, number, capabilities, and limitations of enemy C4I and other information processing systems.
3. Environmental Conditions	3.1 Physical	Factors arising from nature and the physical environment as modified by man. Includes land, sea, air, and space.
	3.1.1 Land	General characteristics of natural and synthetic terrain and geological features. Includes information on buildings and infrastructure (roads, communications, etc.) appropriate to the mission.
	3.1.2 Sea	General characteristics of the ocean surface and subsurface, harbors, and littoral waters.
	3.1.3 Air	General characteristics of the lower atmosphere, including climate, visibility, and atmospheric weapon effects.
	3.1.4 Space	General characteristics of the upper reaches of Earth's atmosphere, including natural (e.g., sunspots) and synthetic (objects in space).
	3.2 Civil	Information about political, cultural, and economic conditions in the areas (hostile, friendly, and neutral) of military interest.
	3.2.1 Political	Those factors that derive from the people, their national government, and international and nongovernment organizations that support or oppose the mission.

TABLE D-1 STANDARD WARFIGHTER INFORMATION (CONT)		
Information Type	**Information Category**	**Definition**
	3.2.2 Cultural	Information relating to language, customs, source of laws, and religion.
	3.2.3 Economic	Information relating to manpower, materiel and money.
4. Situational Information (Friendly and enemy forces information which is specific to a situation.)	4.1 Mission Information	Those factors that frame and influence the execution of the mission. Includes instructions and policies; rules of engagement; status of preparations for the mission; description of the theater; and time constraints.
	4.2 Command, Control, and Communications Conditions	Command relationships and procedures for effective management of forces and accomplishment of the mission. Includes communications systems connectivity and interoperability.
	4.3 Intelligence	Threat-related information and general information regarding the enemy that affects mission accomplishment. Includes enemy doctrine and probable courses of action, and enemy vulnerabilities.
	4.4 Targeting	Information relating to targets. Includes dispersion, camouflage, hardness, identification, mobility, range from potential attacking forces, etc.
	4.5 Deployment, Movement, and Maneuver	Status of lines of communication and planning for deployment, movement or maneuver.
	4.6 Protection	Information regarding rear area security, space control, and air, maritime, and land superiority.
	4.7 Sustainment	Information relating to the sustainment of forces in conducting the mission.

This Page Intentionally Left Blank

ANNEX E

SAMPLE ARCHITECTURE PRODUCTS

The goal is to have at least three examples of each architecture product provided in this Annex. For Version 1.0 only a few examples were readily available.

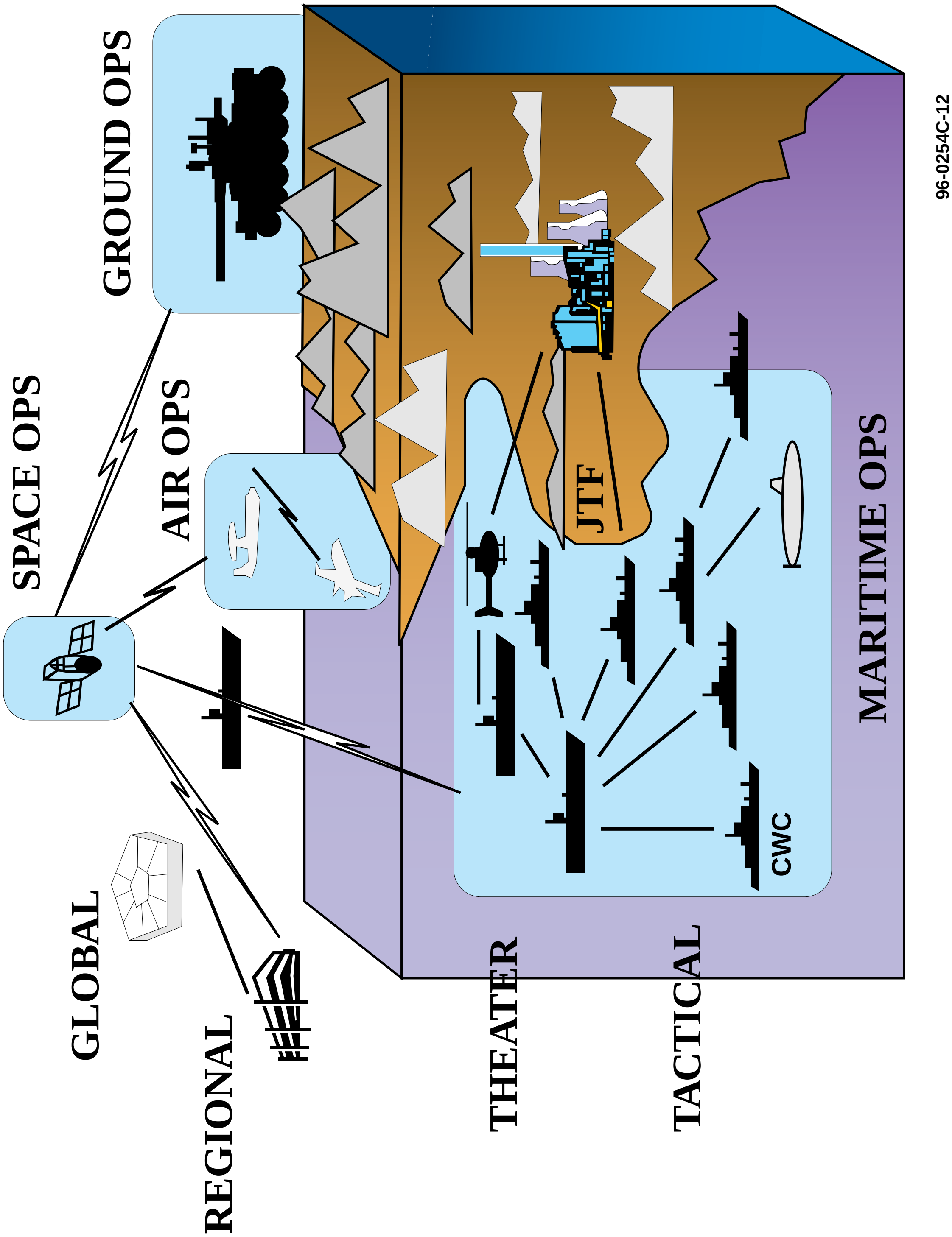


Figure E-1: Operational Concept Diagram

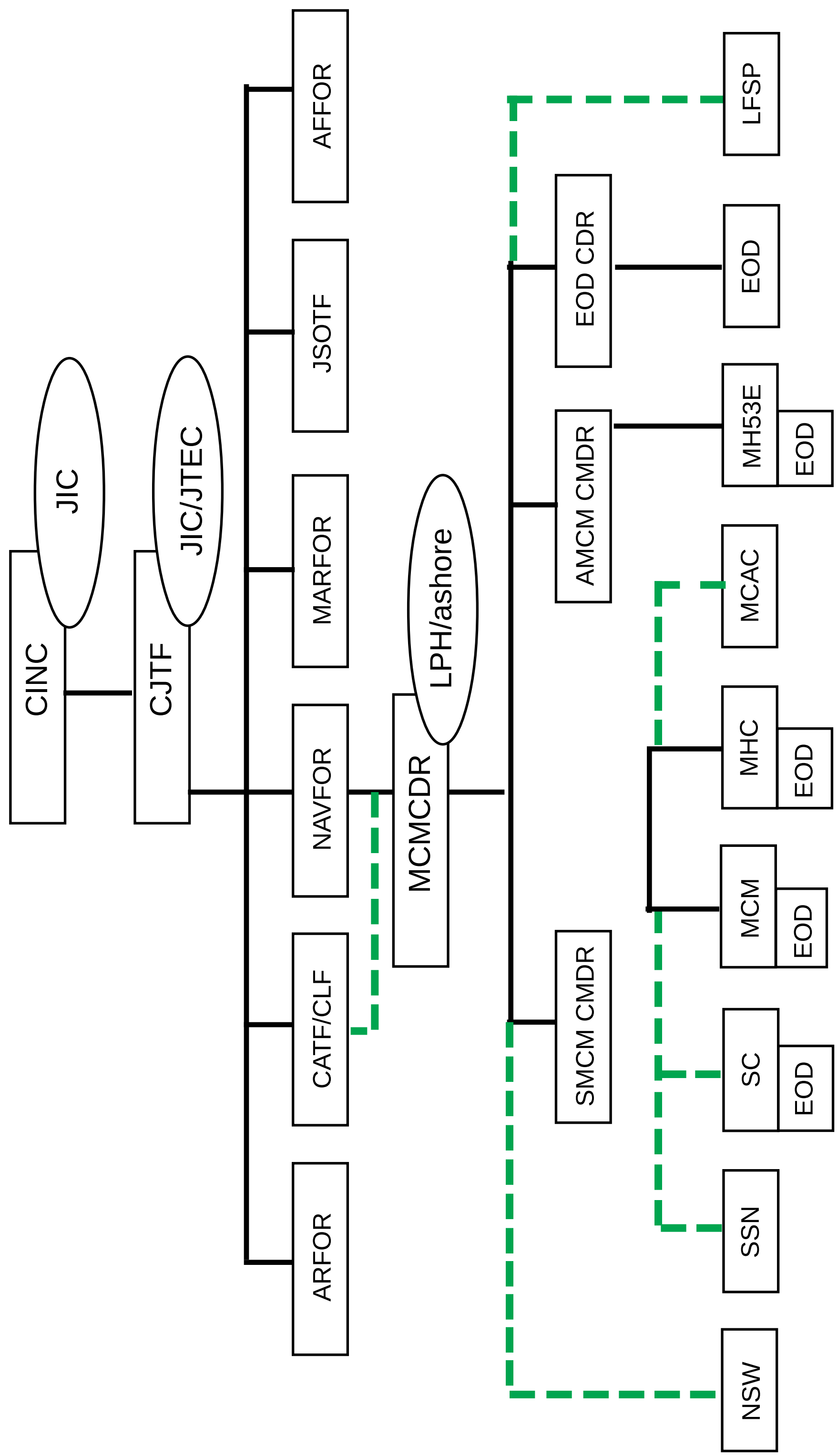
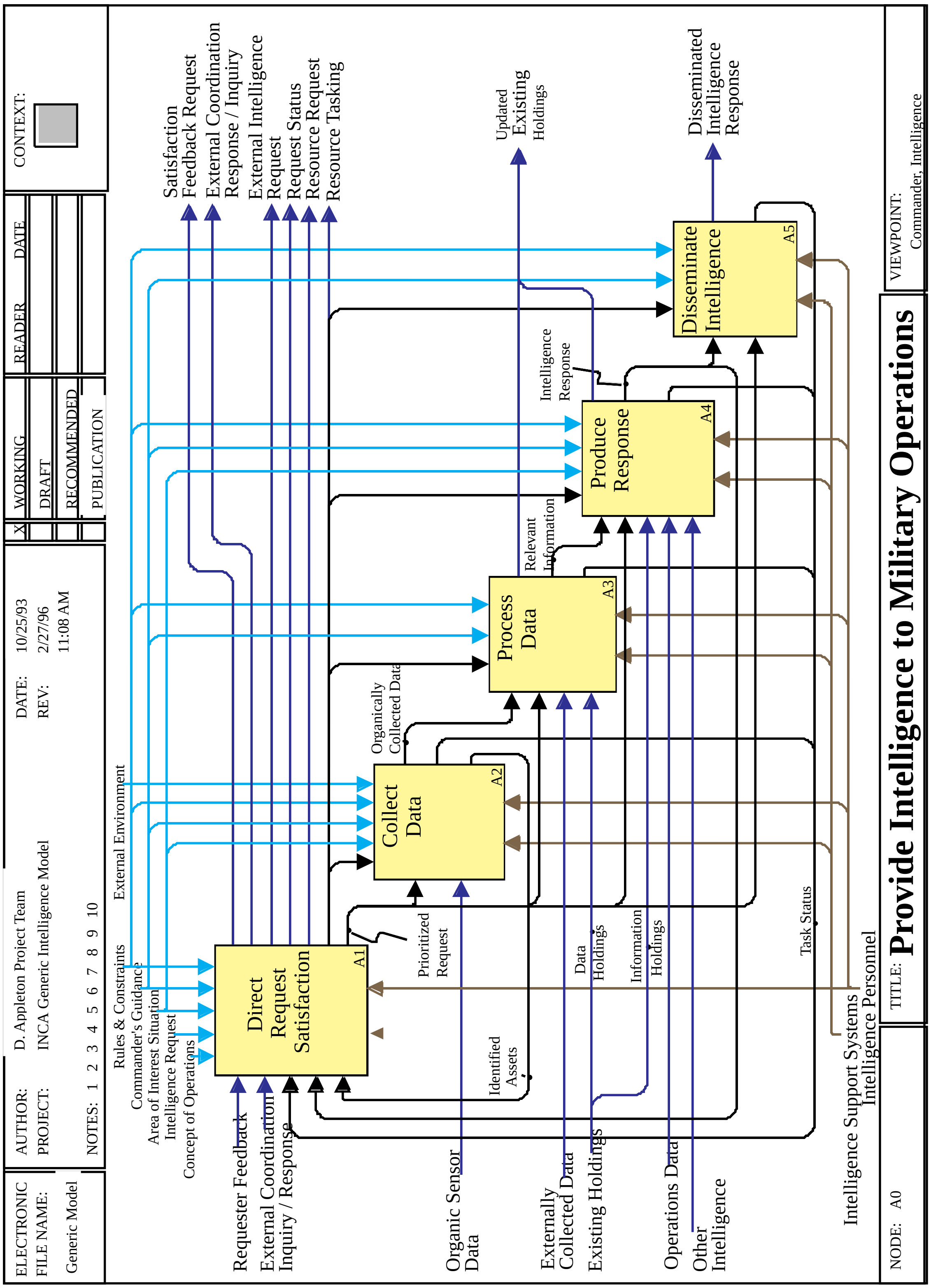


Figure E-2: Command Structure Chart



ELECTRONIC FILE NAME: Generic Model
AUTHOR: D. Appleton Project Team
PROJECT: INCA Generic Intelligence Model
DATE: 10/25/93
REV: 2/27/96 11:08 AM
NOTES: 1 2 3 4 5 6 7 8 9 10
X WORKING
DRAFT
RECOMMENDED
PUBLICATION
READER
DATE
CONTEXT:
Direct Request Satisfaction
A1
Collect Data
A2
Process Data
A3
Produce Response
A4
Disseminate Intelligence
A5
Rules & Constraints
Commander's Guidance
Area of Interest Situation
Intelligence Request
Concept of Operations
External Environment
Satisfaction Feedback Request
External Coordination Response / Inquiry
External Intelligence Request
Request Status
Resource Request
Resource Tasking
Updated Existing Holdings
Disseminated Intelligence Response
Requester Feedback
External Coordination Inquiry / Response
Organic Sensor Data
Externally Collected Data
Existing Holdings
Operations Data
Other Intelligence
Intelligence Support Systems
Intelligence Personnel
Prioritized Request
Identified Assets
Organically Collected Data
Relevant Information
Intelligence Response
Data Holdings
Information Holdings
Task Status
NODE: A0
TITLE: Provide Intelligence to Military Operations
VIEWPOINT: Commander, Intelligence

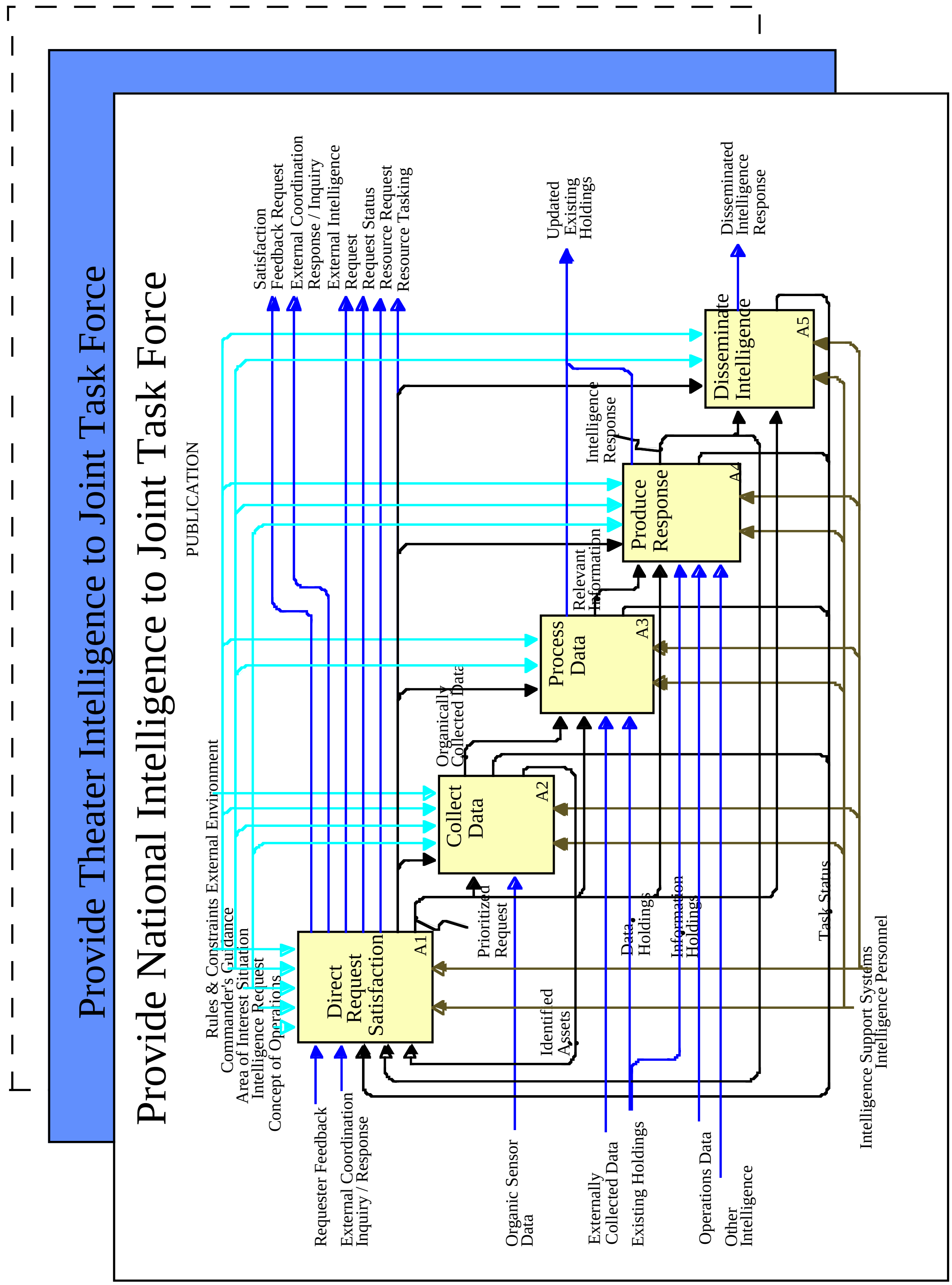


Figure E- 4: Activity Diagrams—Family of Related ModelSs From Different Viewpoints

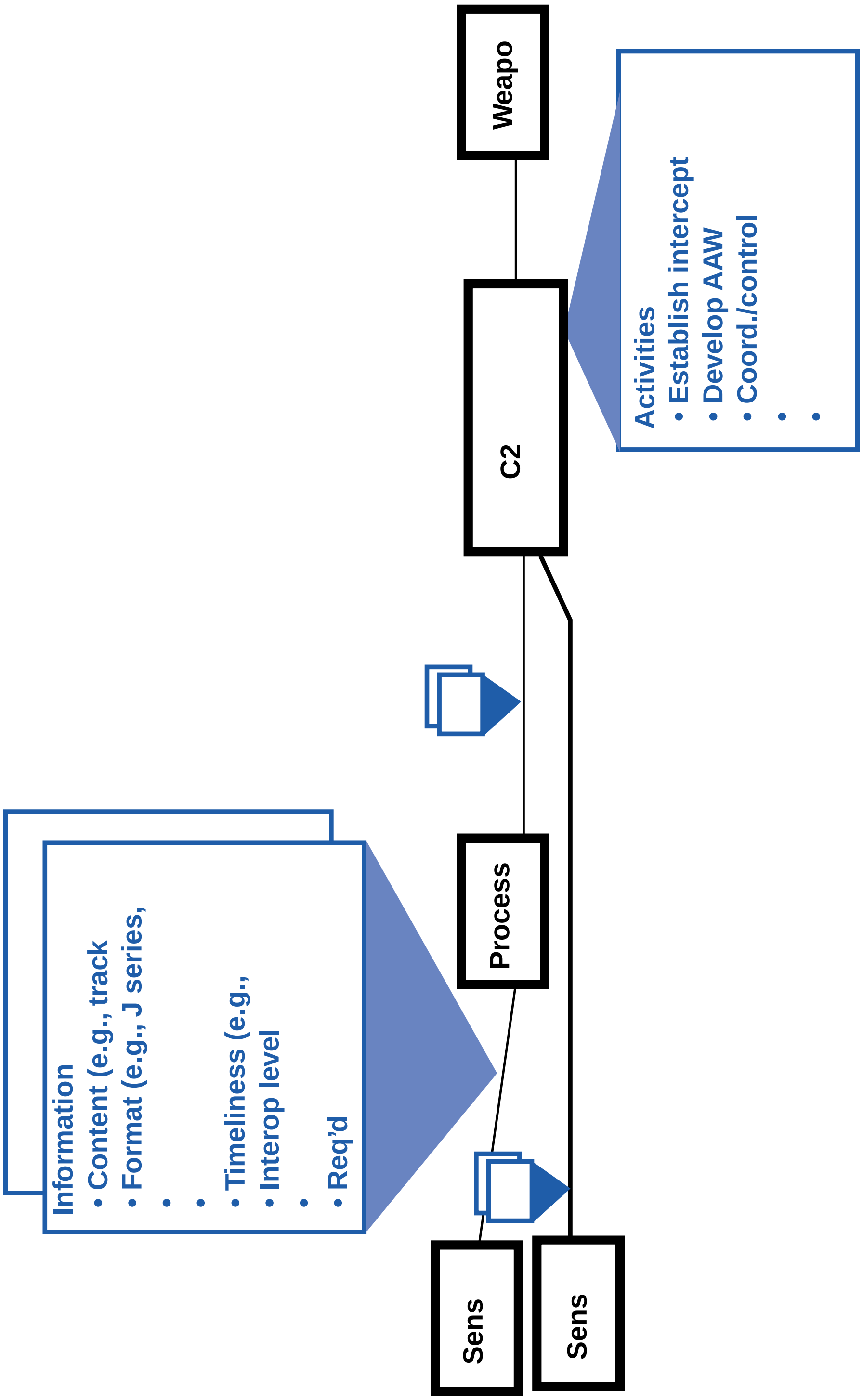


Figure E-5: Basic Node Interoperability Model

Figure E-5: Basic Node Interoperability Model

Supported Operational	Operational Elements		Description of	Required Information Exchange			
Specific Missions, Functions,	**Information Consuming**	**Information Producing**	**Element of Warfighter**	**Type (text, video, voice,**	**Type (text, video, voice,**	**Type (text, video, voice,**	**Type (text, video, voice,**
Provide Guidance	JTF	CINC	Guidance	Text	Secret, Pass W/in	3 MTS Free Text	Message Center
Integrate Intelligence	JTF	DIA	Threat	Text	Secret, Pass every	4 MTS INTSUM	Message Center
Communicate Course	All	JTF	Guidance	Text	Secret, Pass W/in	3 MTS Free Text	Message Center
Assign Aircraft to	Component	JTF	Air Tasking	Text	Secret, Pass every	1 MTS	CTAPS
Receive Weapons	Component	Component	Engagement	Data	Secret, pass w/in	Link 16	Tactical
Process Targets	Component	Component	Engagement	Data	Link 16	Link 16	Tactical
Engage Target	Component	Component	Target	Data	Engagement Qual Data avail at Max Engagement	Updates every	Dedicated

Figure E-6: Sample Information Exchange Matrix

Node	DIA	CINC	JTF	Component A	Component B	Component C	. . N
DIA	Capability — Secret MTS Msg Generation. Capacity — Every 6 Hrs.	NA	Assess to Message Center	NA	NA	NA	NA
CINC	NA	Capability — Secret MTS Msg Generation. Capacity — 3 Msg/ Day W/in 20 min.	Assess to Message Center	NA	NA	NA	NA
JTF	NA	NA	Capability — Generate & Rec Secret MTS Msg Capacity — Send 4/Day Rec 7/Day	Msg Ctr & CTAPS Access	Access To Message Center	Access To Message Center	NA
Component A	NA	NA	NA	Capability — Link 16 Btw C3 & Aircraft, Rec Mts Msg, & Fwd Engagement Data	NA	NA	NA
Component B	NA	NA	NA	Link 16 & Dedicated Path for Sensor Data	Capability — Rec MTS Msgs, Participate in & Display Link 16 Info, & pass engagement data	NA	NA
Component C	NA	NA	NA	NA	NA	Capability — Rec MTS Msgs	NA
. . . N	NA	NA	NA	NA	NA	NA	

Figure E-7: Sample Required Capabilities Matrix

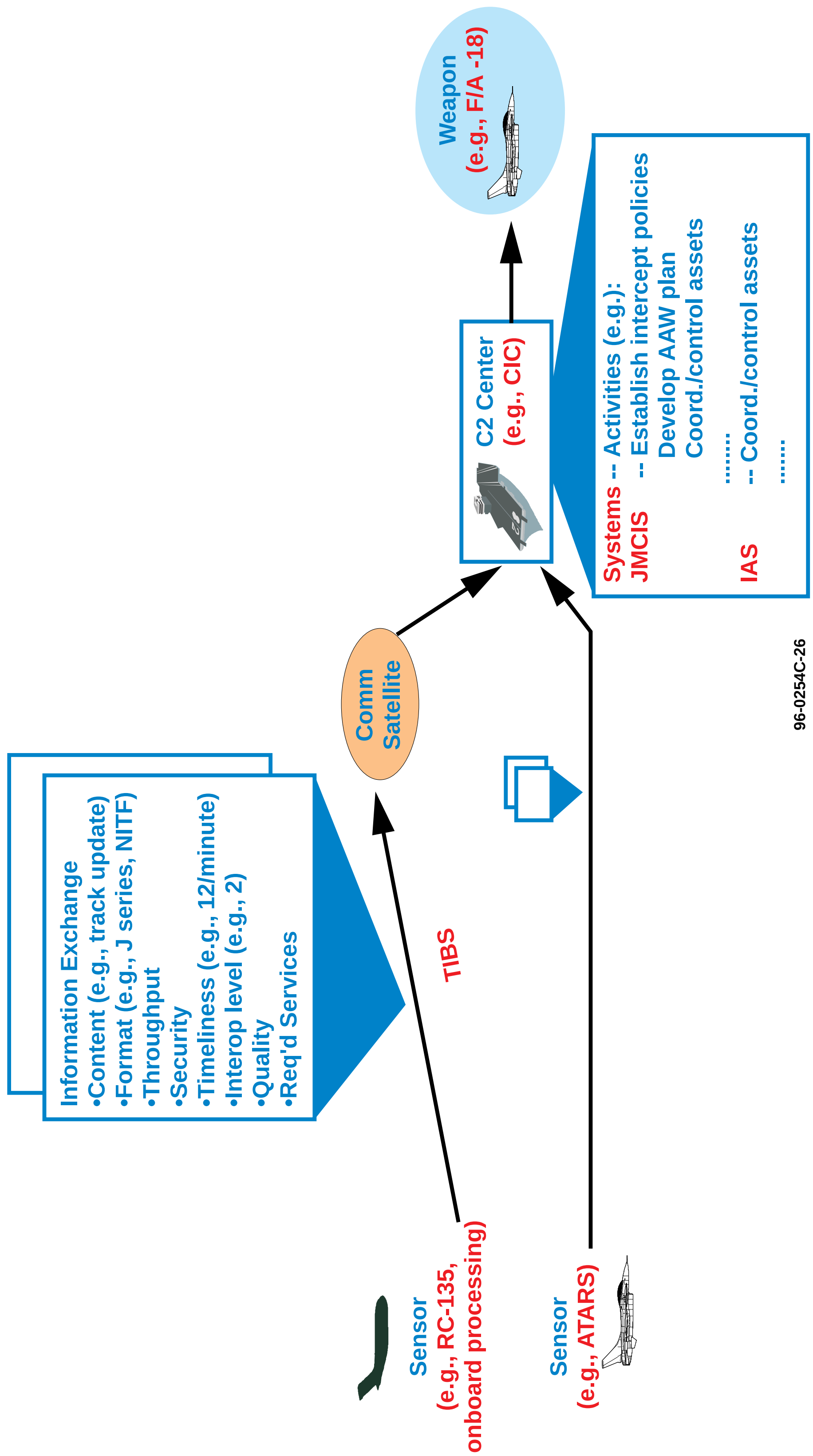


Figure E-8: Systems Overlay on Node Connectivity Model

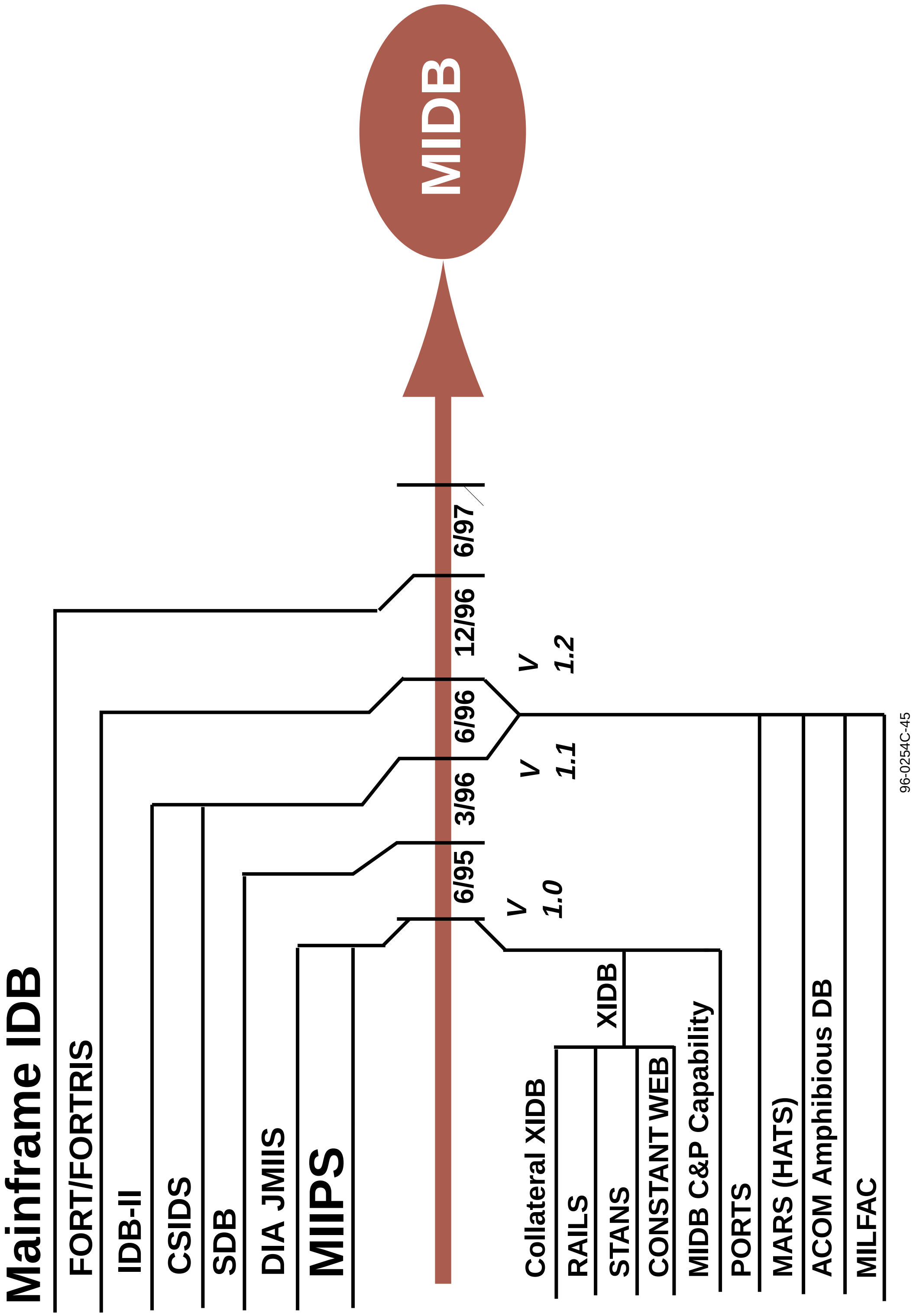
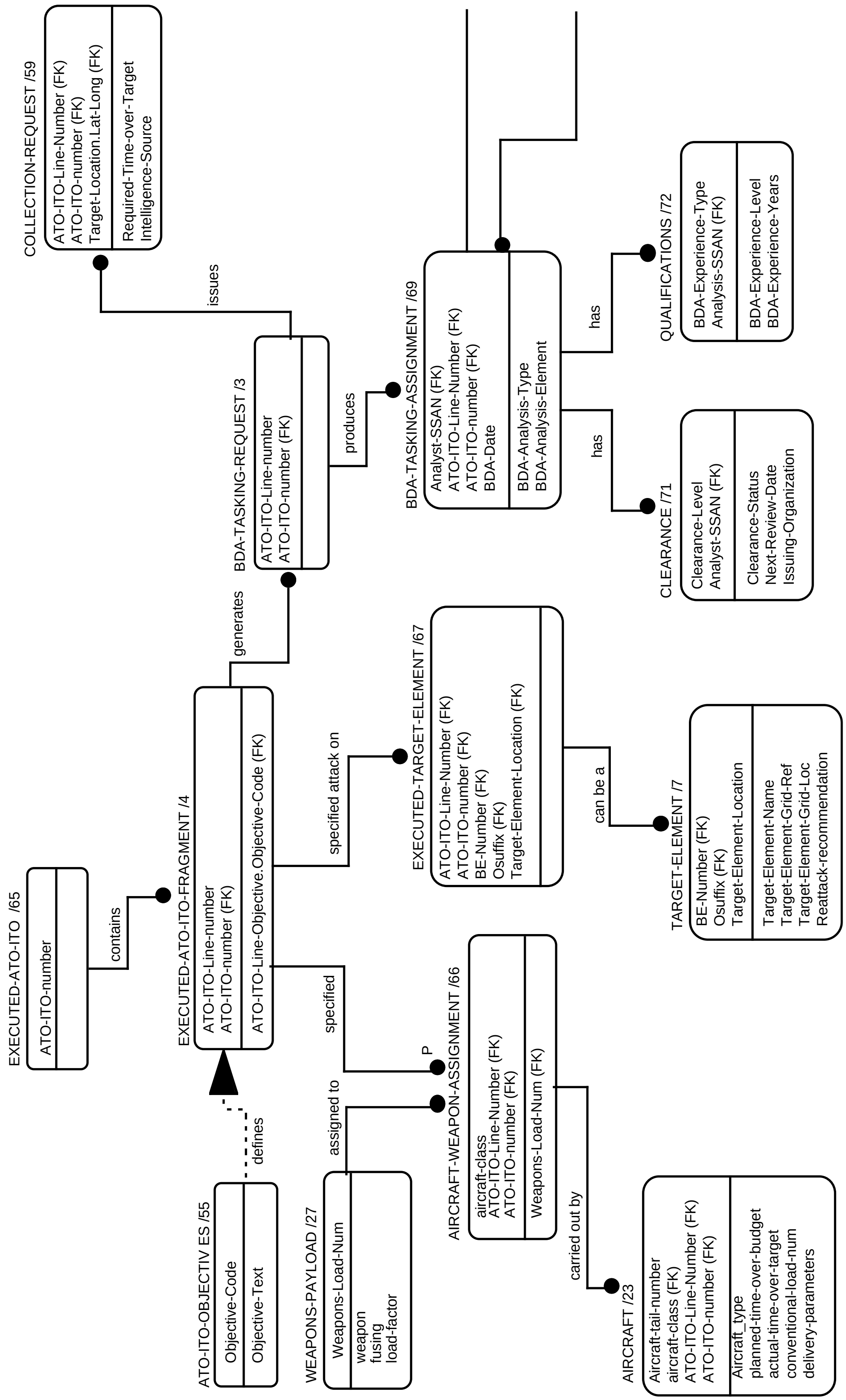


Figure E-9: System Evolution Diagram


COLLECTION-REQUEST /59
ATO-ITO-Line-Number (FK)
ATO-ITO-number (FK)
Target-Location.Lat-Long (FK)
Required-Time-over-Target
Intelligence-Source
issues
BDA-TASKING-REQUEST /3
ATO-ITO-Line-number
ATO-ITO-number (FK)
produces
BDA-TASKING-ASSIGNMENT /69
Analyst-SSAN (FK)
ATO-ITO-Line-Number (FK)
ATO-ITO-number (FK)
BDA-Date
BDA-Analysis-Type
BDA-Analysis-Element
has
QUALIFICATIONS /72
BDA-Experience-Type
Analysis-SSAN (FK)
BDA-Experience-Level
BDA-Experience-Years
has
CLEARANCE /71
Clearance-Level
Analyst-SSAN (FK)
Clearance-Status
Next-Review-Date
Issuing-Organization
generates
EXECUTED-ATO-ITO /65
ATO-ITO-number
contains
EXECUTED-ATO-ITO-FRAGMENT /4
ATO-ITO-Line-number
ATO-ITO-number (FK)
ATO-ITO-Line-Objective.Objective-Code (FK)
specified attack on
EXECUTED-TARGET-ELEMENT /67
ATO-ITO-Line-Number (FK)
ATO-ITO-number (FK)
BE-Number (FK)
Osuffix (FK)
Target-Element-Location (FK)
can be a
TARGET-ELEMENT /7
BE-Number (FK)
Osuffix (FK)
Target-Element-Location
Target-Element-Name
Target-Element-Grid-Ref
Target-Element-Grid-Loc
Reattack-recommendation
specified
P
AIRCRAFT-WEAPON-ASSIGNMENT /66
aircraft-class
ATO-ITO-Line-Number (FK)
ATO-ITO-number (FK)
Weapons-Load-Num (FK)
assigned to
defines
ATO-ITO-OBJECTIV ES /55
Objective-Code
Objective-Text
WEAPONS-PAYLOAD /27
Weapons-Load-Num
weapon
fusing
load-factor
carried out by
AIRCRAFT /23
Aircraft-tail-number
aircraft-class (FK)
ATO-ITO-Line-Number (FK)
ATO-ITO-number (FK)
Aircraft_type
planned-time-over-budget
actual-time-over-target
conventional-load-num
delivery-parameters

This Page Intentionally Left Blank

APPENDIX F

LIST OF ACRONYMS

ACRONYMS

A

AATS	Automated Architecture Tool Suite
AMWG	Architecture Methodology Working Group
ASD (C3I)	Assistant Secretary of Defense for Command, Control, Communications, and Intelligence

C

C/S/As	Commanders in Chief, the military Services, and the DoD agencies
C2	Command and Control
C4I	Command, Control, Communications, Computers, and Intelligence
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CINCs	Commanders in Chief
CISA	C4I Integration Support Activity
CJCS	Chairman of the Joint Chiefs of Staff
COTS	Commercial off-the-shelf
CTT	Commander’s Tactical Terminal

D

DDRS	Defense Data Repository System
DDS	Defense Dissemination System
DISA	Defense Information Systems Agency
DoD	Department of Defense
DoDIIS	Department of Defense Intelligence Information System

DSMSL	Document Style Semantics and Specification Language

E

E-Mail	Electronic Mail
E-R	Entity-Relationship

F

FPI	Functional Process Improvement

G

GIF	Graphic Information File
GPF	Ground Processing Facility
GSM	Ground Station Module
GW	Graphics Workstation

I

IDEF	Integration Definition for Function modeling
IERs	Information Exchange Requirements
ITF	Integration Task Force

J

JICs	Joint Intelligence Centers
JMETL	Joint Mission Essential Task List
JTA	Joint Technical Architecture

L

LAN	Local Area Network

M

MAN	Metropolitan Area Network
MNS	Mission Need Statements
MOP	Memorandum of Policy

N

NMETL	Navy Mission Essential Task List

O

OE	Operating Environment
ORD	Operational Requirements Documents

S

SAASE	Standard Data Element-Based Automated Architecture Support Environment
SBA	Standards Based Architecture

T

TACELINT	Tactical Electronic Intelligence
TACREP	Tactical Report
TAFIM	Technical Architecture Framework for Information Management
TAGS	Tactical Ground Systems
TDDS	Tactical Data Distribution System
TIBS	Tactical Information Broadcast Services
TRAP	Tactical Recovery of Aircraft and Personnel
TRIXS	Tactical Reconnaissance Intelligence Exchange

TRM Technical Reference Model

U

UJTL Universal Joint Task List

W

WAN Wide Area Network